

The R logo is a large, stylized, 3D graphic in the background. It is composed of several blue and yellow rectangular blocks arranged to form the letter 'R'.

# ANÁLISIS DE DATOS CON R



contáctenos: [enei@inei.gob.pe](mailto:enei@inei.gob.pe) / 433-3127



Pasaje Hernán Velarde 285 Lima.

Entre la cuadra 01 y 02 de la Av. Arequipa.

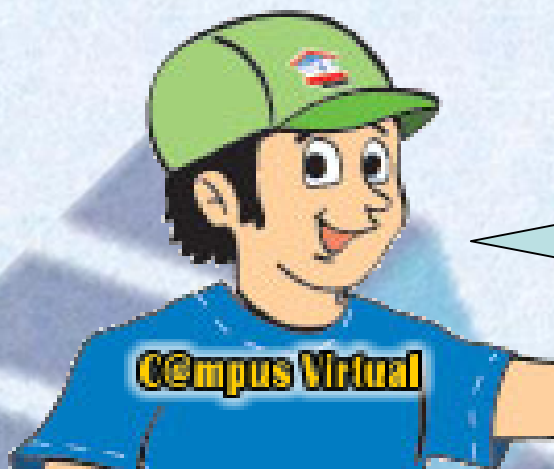
Correo: [enei@inei.gob.pe](mailto:enei@inei.gob.pe) / [campusvirtual@inei.gob.pe](mailto:campusvirtual@inei.gob.pe)

Teléfonos: 433-3127 - 332-4650

## Centro Andino de Formación y Capacitación en Estadística

Cursos Especializados en Estadística e Informática

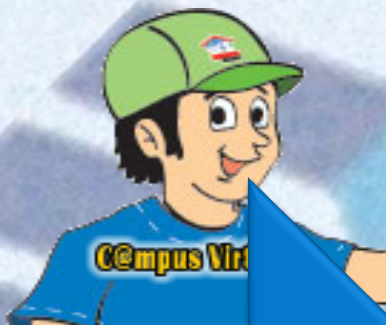




**Estimado alumno,  
buen día. Cualquier  
consulta no dudes en  
comentarme o  
avisarme.**

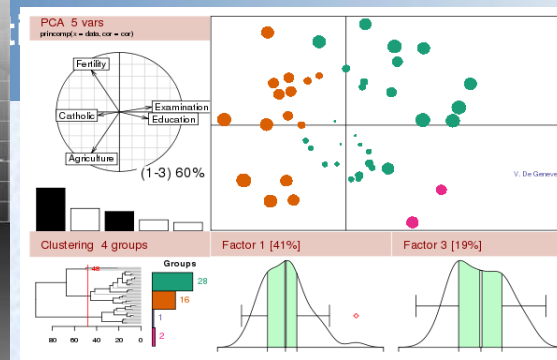
**Este curso es  
netamente práctico y  
se que lograremos  
objetivos importantes.**

**Recuerda siempre nuestro correo  
para cualquier consulta**



**campusvirtual@inei.gob.pe**





# LISTAS Y DATAFRAMES

- Listas
- Data frames
- Trabajando con data frames: attach y detach
- Factores
- Ejercicios propuestos





# Listas (1)

Una lista es un objeto que consiste en una colección ordenada de objetos, llamados componentes. Estos componentes no tienen que ser necesariamente del mismo tipo o *mode*, y pueden ser vectores, *arrays* o nuevas listas. Las listas se construyen con la función *list*.

```
# Creamos una lista
> lst <- list(padre="Carlos", madre="María", num.hijos=3, edad.hijos=c(4,7,9))
> lst
$padre
[1] "Carlos"

$madre
[1] "María"

$num.hijos
[1] 3

$edad.hijos
[1] 4 7 9

# Los elementos de una lista siempre estan numerados, y pueden ser accedidos mediante esos
# números. En este caso los elementos de la lista se indexan mediante dobles corchetes ([[ ]]). Si lst[[4]] es
# un vector, entonces lst[[4]][1] es su primer elemento.
> lst[[1]]
[1] "Carlos"

> lst[[3]]
[1] 3

> lst[[4]][1]
[1] 4
```

# Listas (2)

La función *length*, aplicada a una lista, devuelve el número de componentes "de primer nivel" que contiene.

```
> length(lst)
[1] 4
```

Los componentes de las listas también pueden tener nombres. En este caso, también nos podremos referir a ellos por su nombre además por su posición, ayudados del símbolo del dólar (\$). Esto es útil para no tener que recordar en qué posición está cada componente de la lista.

```
> lst$madre #Equivalente a lst[[2]]
[1] "María"
```

```
# Otra manera de hacerlo
> x <- "madre"
> lst[[x]] #Equivalente a lst[["madre"]]
[1] "María"
```

```
> lst$edad.hijos[2] #Equivalente a lst[[4]][2]
[1] 7
```

# Hay que distinguir entre `lst[1]` y `lst[[1]]`. El primer comando devuelve una sublista. Mientras que el segundo # devuelve el primer componente de la lista.

```
> lst[1] #Si la lista tiene nombres, éstos se transfieren a la sublista.
$padre
[1] "Carlos"
```

```
> lst[[1]] #Se devuelve el primer elemento de la lista respetando su tipo.
[1] "Carlos"
```



# Listas (3)

Los nombres de las listas se pueden añadir con posterioridad a la creación de la lista.

```
# Creamos la lista sin nombres
> lst <- list("Carlos", "María", 3, c(4,7,9))
# Le añadimos los nombres
> names(lst) <- c("padre","madre","num.hijos","edad.hijos")
```

```
# Podemos extender la lista de manera sencilla
> lst[5] <- c("Juan", "Pedro", "Ana")
# Le ponemos el nombre al nuevo elemento de la lista
> names(lst)[5] <- "nombre.hijos"
```

```
# Otra manera alternativa de hacerlo
> lst$nombre.hijos <- c("Juan", "Pedro", "Ana")
```

Para concatenar listas utilizaremos la función *c*, de la manera *c(lista1, lista2, ..., lista n)*.

Cuando nos referimos a los elementos de las listas por sus nombres podemos utilizar el número mínimo de letras que indentifique de manera única a los diferentes componentes. De esta manera, *lst\$padre* podría ser especificado como *lst\$p*.

# Data frames

Un *data frame* es una lista de variables que tienen la misma longitud y que han de estar identificadas por un nombre único. Se crean mediante la función *data.frame*.

```
# Creamos el data frame
> df <- data.frame(Nombre=c("Pedro","María","José","Marta"),
                  Edad=c(27,34,40,39),
                  Poblacion=c("Zaragoza","Madrid","Valencia","Barcelona"),
                  Sexo=c("H","M","H","M"),
                  Casado=c(F,T,T,F))
> df #Para acceder a una variable en concreto lo haremos con el dólar ($), como en el caso de las listas.
```

	Nombre	Edad	Poblacion	Sexo	Casado
1	Pedro	27	Zaragoza	H	FALSE
2	María	34	Madrid	M	TRUE
3	José	40	Valencia	H	TRUE
4	Marta	39	Barcelona	M	FALSE

En este caso todas las columnas del *data frame* han de tener 4 elementos o un número múltiplo de 4. Aplicando el reciclaje, R extiende las columnas más cortas para igualarlas en longitud respecto a la más larga.

Las matrices y las listas también pueden ser transformadas a *data.frames*. En este caso cada columna de la matriz o elemento de la lista pasan a ser una columna del nuevo *data frame*.

Los *data frames* se extienden igual que las listas (*df\$nueva.col <- valores*)



# Trabajando con data frames: attach y detach

A veces puede ser engorroso trabajar con *data frames*, debido a que para acceder a las variables hemos de escribir el nombre del *data frame* seguido de un dólar y del nombre de la variable. R nos permite hacer visible el contenido de un *data frame* o de una lista, de manera que no hemos de escribir su nombre cada vez que queremos acceder a una de sus variables. Esto lo hacemos con la instrucción *attach*. Para volver a hacer "invisible" el *data frame* utilizaremos *detach*.

```
# Hacemos el attach del data frame
```

```
> attach(df)
```

```
# Vemos que el data frame ya está cargado en nuestro entorno.
```

```
> search()
```

```
[1] ".GlobalEnv"      "df"               "package:methods" "package:stats"    "package:graphics" "package:utils"
```

```
[7] "Autoloads"       "package:base"
```

```
> ls(2)
```

```
[1] "Casado"  "Edad"    "Nombre"  "Poblacion" "Sexo"
```

```
# Accedemos directamente a las variables del data frame
```

```
> Nombre #En este caso la variable ha sido transformada a factor
```

```
[1] Pedro María José Marta
```

```
Levels: José María Marta Pedro
```

```
# Dejamos de hacer "visible" el data frame con la instrucción detach.
```

```
> detach(df)
```

```
> search()
```

```
[1] ".GlobalEnv"      "package:methods" "package:stats"    "package:graphics" "package:utils"    "Autoloads"
```

```
[7] "package:base"
```

# Factores

R dispone de un tipo especial de datos llamado factor. Los vectores de factores sirven principalmente para agrupar elementos de otros vectores de la misma longitud. Los factores se utilizan principalmente en fórmulas para realizar modelos.

```
> df$Sexo  
[1] H M H M  
Levels: H M
```

Los *levels* contienen las distintas etiquetas. A cada etiqueta diferente R le asigna un número identificador por orden alfabético. De esta manera, podríamos redefinir los factores como etiquetas de texto que tienen asociado un código interno.

```
# Para ver los levels de una variable factor utilizaremos la instrucción levels.  
> levels(df$Sexo)  
[1] "H" "M"
```

```
# Si queremos reordenarlos utilizaremos la instrucción relevel  
> relevel(df$Sexo,2) #Le decimos que el 2º factor pase a la 1ª posición.  
[1] H M H M  
Levels: M H
```

```
# Para crear una variable factor, R dispone de la instrucción factor, a la que se le suele pasar un vector de  
# caracteres  
> a <- factor(c("a","b","c","b","a"))  
> a  
[1] a b c b a  
Levels: a b c
```



# Ejercicios propuestos

1. Crear una lista donde guardaremos información de 3 personas. La lista tendrá 4 componentes, que serán vectores: nombre (carácter), edad (numérico), sexo (factor, H – M), y una variable booleana que indica si esa persona ya trabaja.
2. Crear un *data frame* que guarde la misma información que la lista anterior. Primero crearlo partiendo de cero y después aprovechar que ya tienes la lista para crearlo de una manera más directa. ¿Hay problemas al transformar un objeto *list* a un objeto *data frame*? ¿Hace falta especificar que el sexo es una variable factor? ¿De qué tipo es la variable "nombre"? Con la ayuda de la función *as.character* transformarla en un vector de caracteres.
3. Añadirle un nuevo vector a la lista que indique la profesión de cada persona. Hacer lo mismo con el *data frame*. Hacerlo todo sin rehacer ni la lista ni el *data frame* de nuevo. Asegurar que la nueva información tenga su *name* correspondiente.
4. Hacer un *attach* del *data frame*. Comprobar que sus columnas son accesibles sin hacer mención explícita al propio *data.frame*. Descargarlo con *detach*.
5. Practicar un poco con la lista y el *data.frame* que tienes creados. ¿Puedes acceder al *data frame* con índices numéricos como si fuera una matriz? ¿Sabes hacer una subselección del *data frame*? ¿Y de la lista?

## Comunicación constante con la Escuela del INEI

**Correo de la Escuela del INEI**  
**[enei@inei.gob.pe](mailto:enei@inei.gob.pe)**

**Área de Educación Virtual**  
**[campusvirtual@inei.gob.pe](mailto:campusvirtual@inei.gob.pe)**

