

## GUÍA PARA EL USO DEL PAQUETE ESTADÍSTICO R

### INSTALACIÓN

R es un paquete estadístico que trabaja bajo línea de comandos pero existe una interficie gráfica, el R-Commander, que facilita su uso. Para su instalación, se puede visitar el siguiente enlace para descargar el ejecutable encargado de la instalación:

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

También es un referente <http://knuth.uca.es/R>

Una vez descargado se procederá a su instalación:

Linux(Ubuntu):

[https://www.youtube.com/watch?v=T\\_0LbTxvs5s](https://www.youtube.com/watch?v=T_0LbTxvs5s)

Windows:

<https://www.youtube.com/watch?v=MFfRQuQKGYg>

### DIRECTORIO DE TRABAJO

R utiliza un directorio de trabajo por defecto, en este caso mis documentos, para detectar cual es éste y donde se encuentra, existe la orden getwd() (Get Working Directory).

Este directorio de trabajo puede cambiarse con setwd("C:/RUTA") o setwd("/home/...").

\*Es necesario poner el path completo del archivo si éste no se encuentra en el directorio de trabajo.

### CARGA DE DATOS

Cabe destacar que se puede usar cualquier gestor de cálculo gratuito capaz de generar ficheros ".csv", los cuales separan los datos por ",". Se ha decidido usar archivos CSV, porque se consideran genéricos en casi todas las suites de ofimática.

### DESDE EXCEL

Los siguientes pasos son para importar ficheros desde excel.

1. Se procede a guardar el archivo .csv en mis documentos.

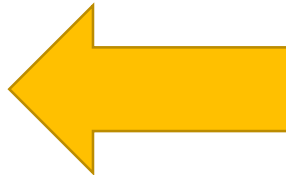
Generalmente la ruta es: C:\Users\TUUSUARIO\Documents\TUARCHIVO.csv

2. Cargar el archivo con el nombre temporal, por ejemplo.

```
$ temporal <- read.csv2("TUARCHIVO.csv")
```

```
> temporal = read.csv2("Pr3es11.csv", header = T)
> temporal
```

```
  Total    F    NF
1  0.97 0.48 0.97
2  0.72 0.71 0.72
3  1.00 0.98 1.00
4  0.81 0.68 0.81
5  0.62 1.18 0.62
6  1.32 1.36 1.32
7  1.24 0.78 1.24
8  0.99 1.64 0.99
9  0.90  NA  0.90
10 0.74  NA  0.74
11 0.88  NA  0.88
12 0.94  NA  0.94
13 1.16  NA  1.16
```



	A	B	C
1	Total	F	NF
2	0,97	0,48	0,97
3	0,72	0,71	0,72
4	1	0,98	1
5	0,81	0,68	0,81
6	0,62	1,18	0,62
7	1,32	1,36	1,32
8	1,24	0,78	1,24
9	0,99	1,64	0,99
10	0,9		0,9
11	0,74		0,74
12	0,88		0,88
13	0,94		0,94

3. Verificar la correcta carga

```
$ View(temporal)
```

## EXPORTAR DATOS

Para exportar los datos se va a proceder a realizar un proceso inverso al anterior. Cabe destacar que debido a la utilización en este caso del Excel con el formato Español, las comas y los puntos están invertidos por lo tanto para una correcta carga se debe usar el siguiente comando:

- `write.csv2(A,"fichero.csv", row.names = FALSE)`

De esta forma se consigue cargar los datos sin ningún tipo de dificultad o error.

## FUNCIONES BÁSICAS

Una vez se obtiene el archivo temporal, que se puede decir que es un vector, donde cada elemento del vector es un conjunto de datos, se podría trabajar desde él si cada elemento, es decir cada columna, tuviera el mismo número de datos.

Sino, como es el ejemplo gráfico anterior, se debe copiar cada columna en una nueva variable. Esto es debido a que R estipula que todas las columnas deben tener la misma longitud provocando en caso contrario errores en cálculos, como por ejemplo el de la media.

Para solucionar este inconveniente, se debe usar el siguiente comando:

```
$ F <- temporal$F[!is.na(temporal$F)]
```

F → es la variable destino.

temporal\$F → es el vector de columnas(temporal) con la columna específica(F)

is.na() → Función genérica de R para comprobar si el dato es accesible.

\*Por qué no usar la misma variable (temporal\$F) puesto que ésta tiene una longitud “N”, esta nueva longitud “n”, donde ( $n < N$ ), provoca que duplique datos hasta que se complete todo el vector.

Una vez se obtienen los datos como vectores, sin variables no accesibles, se va a proceder a explicar cómo se calculan las funciones básicas.

## MOSTRAR OBJETOS

```
$ objects()
```

Muestra todos los objetos con los que se está trabajando en esa sesión.

## MOSTRAR CONTENIDO DE OBJETOS

```
$view(Nombre del objeto)
```

Muestra el contenido del objeto por pantalla.

## MEDIA

```
$ media = mean (Tot)
```

Donde Tot, es un conjunto de datos que va desde 1 a n componentes.

## DESVIACIÓN TÍPICA

```
$ desTip = ds(Tot)
```

Donde Tot, es un conjunto de datos que va desde 1 a n componentes.

## GENERACIÓN DE DATOS ALEATORIOS

```
$ rnorm(#datos, mean = media, sd = desviación estándar)
```

La funció `random norm` en este caso genera `n` datos aleatorios donde `n` es el #datos a partir de una media y una desviación estándar introducidas.

## T DE STUDENT

Para la realización de intervalos de confianza mediante la *t* de Student existen varios factores a tener en cuenta. Veamos un ejemplo para dos poblaciones.

```
$ t.test(muestraA,muestraB, alternative = 'two.sided', conf.level=.99, paired = TRUE)
```

`muestraA` y `muestraB`, son las dos muestras de las que se dispone para realizar el intervalo de confianza. El elemento `alternative` es el que indica el tipo de hipótesis alternativa, dando opción a:

- `two.sided`
- `less`
- `greater`

`conf.level` es el nivel de confianza con el que se quiere asegurar que la hipótesis nula se aceptada o rechazada y por último `paired` indica si las dos muestras, en este caso `muestraA` y `muestraB` guardan relación entre ellas, es decir, son dependientes.

```
> t.test(A,B)

Welch Two Sample t-test

data:  A and B
t = 2.7232, df = 137.799, p-value = 0.007302
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.0637696 0.4018875
sample estimates:
mean of x mean of y
 1.593047  1.360219
```

## ANOVA

Para determinadas funciones o cálculos, es útil exportar los datos para obtener una mayor facilidad de trabajo y volver a importarlos una vez manipulados como nos interese.

Por ejemplo, consideremos que se tienen tres muestras con las que se quiere realizar un contraste ANOVA de un factor, donde el factor es el tipo de sistema que trabajas, y estos datos se tiene divididos por el tipo de muestra, es decir, como los que siguen.

A

```
> A
[1] 1.7317686 1.6765955 2.0138383 0.4757245 1.7987050 2.1413851 1.2056970
[8] 1.8092572 2.0481029 0.9515876 1.3488882 1.2510428 1.1296388 1.5118535
[15] 1.3455583 1.0396647 2.9522062 1.3249790 1.9071859 1.3583652 1.5413668
[22] 2.2011603 2.4200877 1.7762684 2.0248086 2.3895003 1.7441873 1.5711805
[29] 2.3423356 1.3411519 1.2532538 0.7449063 2.3728734 1.5433987 1.3912158
[36] 1.9130104 1.8795851 1.2455141 1.6633779 0.9883534 1.3182204 2.6692721
[43] 1.2766803 1.0964122 1.4562750 0.6366550 1.8177914 1.7451732 1.6848500
[50] 2.1886604 1.7295155 1.2325554 1.1679798 1.0157907 1.0507630 1.6535560
[57] 2.0120621 1.6082586 1.1187826 1.0917528 1.2833327 1.3869191 0.6066325
[64] 1.9754344 1.8826503 2.1172384 1.6996135 2.3526039 1.7745813 1.4937014
```

B

```
> B
[1] 0.9623808 1.3719786 0.9892856 1.2036644 0.3340141 1.4233128 1.5709396
[8] 1.2336167 1.5394881 1.2042335 0.9250564 1.2924074 1.5530904 1.1242043
[15] 1.3914390 1.3464719 1.1428164 2.2542503 1.1087669 0.8241873 2.5821937
[22] 2.0595860 0.2436752 0.8285381 1.8583737 1.5868668 0.5085534 2.2006143
[29] 2.0212306 1.7160205 2.1757467 0.6477795 1.0056379 1.0781664 0.5574438
[36] 1.6893156 1.3893762 0.9814747 1.0125848 1.0149274 0.9352079 1.2242496
[43] 0.6135233 2.0981863 1.1192410 1.6373404 1.8593394 1.9198147 1.4256035
[50] 0.5687137 1.4262492 0.8856020 1.9774536 2.4002835 1.0264333 1.3538098
[57] 1.9526594 1.3997391 1.1804711 1.9022044 1.5627002 1.7250645 1.1515168
[64] 1.6308331 0.8352442 2.4666969 1.0504278 1.4005688 1.1992015 1.3332084
```

C

```
> C
[1] 1.5941001 1.3007393 1.1751227 1.2820279 1.6300837 1.1277791 0.4442057
[8] 1.7569830 0.7060949 1.5622815 2.1192295 1.5037038 1.3121400 2.2188765
[15] 0.5103945 1.6105856 0.4939159 1.3631754 1.0919321 1.7445517 2.8468400
[22] 0.7823770 1.7946235 1.0108857 1.9514035 1.5782186 1.4654297 2.5008519
[29] 2.0775632 1.2330794 1.4803446 1.2539340 0.7595254 2.0807252 1.3032827
[36] 1.6026642 1.1066919 2.4551267 1.2377392 2.4640019 2.5370054 0.9490731
[43] 1.4868853 1.5701708 2.1526256 0.8288630 0.8812878 1.5417385 2.0773052
[50] 0.7360070 2.0093333 1.6643516 1.3528570 1.4900925 1.3581164 1.1142405
[57] 1.9222403 1.9989558 0.5761189 1.7826691 0.5724514 1.6180593 1.0698408
[64] 1.0111279 1.0425672 0.8116885 1.1724430 1.7545110 2.3322998 0.7981888
```

Primero se deben juntar estos datos en un solo conjunto, para poder posteriormente exportarlos a una plantilla csv sin ningún tipo de problemas. Para ello se va a proceder a usar data.frames, que como su nombre indica son ventanas de datos.

```
> conjunt <- data.frame(A,B,C)
```

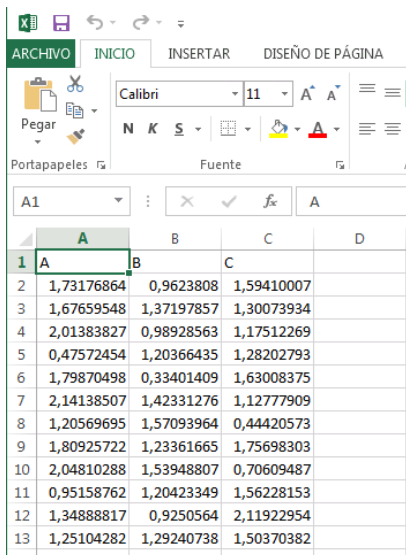
Para realizar una comprobación del contenido de este conjunto creado, simplemente se teclea el nombre de la variable, para poder así visualizarlo en pantalla.

```
> conjunt
      A      B      C
1  1.7317686 0.9623808 1.5941001
2  1.6765955 1.3719786 1.3007393
3  2.0138383 0.9892856 1.1751227
4  0.4757245 1.2036644 1.2820279
5  1.7987050 0.3340141 1.6300837
6  2.1413851 1.4233128 1.1277791
7  1.2056970 1.5709396 0.4442057
8  1.8092572 1.2336167 1.7569830
9  2.0481029 1.5394881 0.7060949
10 0.9515876 1.2042335 1.5622815
11 1.3488882 0.9250564 2.1192295
12 1.2510428 1.2924074 1.5037038
13 1.1296388 1.5530904 1.3121400
14 1.5118535 1.1242043 2.2188765
15 1.3455583 1.3914390 0.5103945
16 1.0396647 1.3464719 1.6105856
17 2.9522062 1.1428164 0.4939159
18 1.3249790 2.2542503 1.3631754
19 1.9071859 1.1087669 1.0919321
20 1.3583652 0.8241873 1.7445517
21 1.5413668 2.5821937 2.8468400
22 2.2011603 2.0595860 0.7823770
```

Una vez se tiene el data.frame construido, se va a proceder a exportar como se ha explicado en el apartado de exportación de datos.

```
> write.csv2(conjunt, "anova.csv", row.names = FALSE)
```

- Se usa el comando csv2 debido a la distribución del paquete Excel de office, puesto que en España se usa la “,” como separador decimal, mientras que con otra distribución el encargado de ese trabajo es el “.”.



	A	B	C	D
1	A	B	C	
2	1,73176864	0,9623808	1,59410007	
3	1,67659548	1,37197857	1,30073934	
4	2,01383827	0,98928563	1,17512269	
5	0,47572454	1,20366435	1,28202793	
6	1,79870498	0,33401409	1,63008375	
7	2,14138507	1,42331276	1,12777909	
8	1,20569695	1,57093964	0,44420573	
9	1,80925722	1,23361665	1,75698303	
10	2,04810288	1,53948807	0,70609487	
11	0,95158762	1,20423349	1,56228153	
12	1,34888817	0,9250564	2,11922954	
13	1,25104282	1,29240738	1,50370382	
14	1,12963878	1,55309044	1,31214000	

Una vez se obtienen los datos en excel, se busca juntar estos en una sola columna, colocando en la segunda columna a qué sistema pertenece (1,2 o 3).

ABC	Sistema
1,73176864	1
1,67659548	1
2,01383827	1
0,47572454	1
1,79870498	1
2,14138507	1
1,20569695	1
1,80925722	1
2,04810288	1
0,95158762	1
1,34888817	1
1,25104282	1
1,12963879	1

Ya solo queda exportar los datos a R para realizar el análisis estadístico correctamente. Para exportar estos datos se procede a realizar el siguiente proceso:

```
> valores <- read.csv2("anova.csv")
> valores
```

	ABC	Sistema
1	1.7317686	1
2	1.6765955	1
3	2.0138383	1
4	0.4757245	1
5	1.7987050	1
6	2.1413851	1
7	1.2056970	1
8	1.8092572	1
9	2.0481029	1
10	0.9515876	1
11	1.3488882	1
12	1.2510428	1
13	1.1296388	1
14	1.5118535	1
15	1.3455583	1
16	1.0396647	1
17	2.9522062	1

Como se puede observar, dentro de valores ahora se guardan dos conjuntos de datos, por un lado el conjunto de datos (ABC) y por el otro el indicador del tipo de sistema al que pertenece cada valor.

Solo queda ejecutar la orden para realizar el contraste de ANOVA de un factor, que en este caso es el sistema.

```
> anova <- aov(ABC~Sistema, valores)
> anova
Call:
  aov(formula = ABC ~ Sistema, data = valores)

Terms:
                Sistema Residuals
Sum of Squares   0.68138   57.82505
Deg. of Freedom      1     208

Residual standard error: 0.5272618
Estimated effects may be unbalanced
```

La primera expresión anova es donde se va a guardar el resultado de la operación, siendo así el objeto de interés una vez realizado el cálculo.

Para realizar el contraste se usa la expresión:

```
summary(aov.Sistema)
```

con lo que se obtiene el cuadro típico ANOVA con la información del Estadístico de Prueba y el p-valor correspondiente.