

Chuncks

Imanol

10/2/2021

Chuncks en Python

Podemos darle un nombre a cada chunk. `ctrl + alt + i` para sacar el chunk automaticamente.

Parametro echo

Nos dice si queremos mostrar o no el codigo, por defecto si no lo ponemos esta a `true`.

Vamos a calcular $\sqrt{2} - e^2$:

```
sqrt(2) - exp(-2)
```

```
## [1] 1.278878
```

Vamos a calcular $\sqrt{2} - e^2$:

```
## [1] 1.278878
```

Parametro eval

Nos dice si queremos evaluar o no el codigo (nos da o no el resultado, se ejecuta o no), por defecto si no lo ponemos esta a `true`.

Vamos a calcular $\sqrt{2} - e^2$:

```
sqrt(2) - exp(-2)
```

```
## [1] 1.278878
```

Vamos a calcular $\sqrt{2} - e^2$:

```
sqrt(2) - exp(-2)
```

Parametro message y warning

`message` nos dice si se esta cargando una libreria, por defecto `TRUE`

```
library(magic)
```

```
## Loading required package: abind
```

```
magic(6)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    7    6   35   34   15   14
## [2,]    8    5   33   36   16   13
## [3,]   27   26   19   18   11   10
## [4,]   25   28   20   17    9   12
## [5,]   23   22    3    2   31   30
## [6,]   21   24    1    4   29   32
```

```
library(magic)
```

```
magic(6)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    7    6   35   34   15   14
## [2,]    8    5   33   36   16   13
## [3,]   27   26   19   18   11   10
## [4,]   25   28   20   17    9   12
## [5,]   23   22    3    2   31   30
## [6,]   21   24    1    4   29   32
```

Parametro comment

Nos quita los hastags si le ponemos comment = NA

```
library(magic)
```

```
magic(6)
```

```
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    7    6   35   34   15   14
[2,]    8    5   33   36   16   13
[3,]   27   26   19   18   11   10
[4,]   25   28   20   17    9   12
[5,]   23   22    3    2   31   30
[6,]   21   24    1    4   29   32
```

Parametro results

El parametro hide nos oculta el resultado pero si se evalua

```
library(magic)
```

```
magic(6)
```

El parametro markup nos da el resultado con dos almohadillas (el que esta por defecto).

```
library(magic)
magic(6)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    7    6   35   34   15   14
## [2,]    8    5   33   36   16   13
## [3,]   27   26   19   18   11   10
## [4,]   25   28   20   17    9   12
## [5,]   23   22    3    2   31   30
## [6,]   21   24    1    4   29   32
```

El parametro asis nos da el resultado tal cual en texto.

```
library(magic)
magic(6)
```

```
      [,1] [,2] [,3] [,4] [,5] [,6]

[1,] 7 6 35 34 15 14 [2,] 8 5 33 36 16 13 [3,] 27 26 19 18 11 10 [4,] 25 28 20 17 9 12 [5,] 23 22 3 2 31 30 [6,] 21
24 1 4 29 32
```

El parametro hold nos da los resultados todos al final.

Sin hold

```
sqrt(2) - exp(-2)
```

```
## [1] 1.278878
```

```
x=1:5
sqrt(x)
```

```
## [1] 1.000000 1.414214 1.732051 2.000000 2.236068
```

Con hold

```
sqrt(2) - exp(-2)
x=1:5
sqrt(x)
```

```
## [1] 1.278878
## [1] 1.000000 1.414214 1.732051 2.000000 2.236068
```

Variables en medio de texto

Cuando queremos hacer la raiz cuadrada de dos, podemos hacerlo:

- En L^AT_EX: $\sqrt{2}$
- En R haciendo 1.4142136
- La frase completa: $\sqrt{2} = 1.4142136$