

Notas de Markdown

Carlos, C.

2022-07-31

R Markdown

[Chunk](#). Bloque de código

Los bloques de código de R de un documento de un documento R Markdown se indican de la siguiente manera:

Para una variable aleatoria $X \sim \exp(\lambda)$ siempre se cumple que $F(E[X]) = 1 - e^{-1}$.

```
1-exp(-1)
```

```
## [1] 0.6321206
```

El anterior **chunk** se puede crear con el short cut *Alt + Ctrl + I*.

Para hacer una visualización del Markdown que hemos hechos debemos hacer un **Knit**, se puede hacer dando clics sobre el icono que aparece en RStudio o con el short cut *Ctrl+Shift+K*.

Al momento de definir un chunk se pueden establecer *parámetros*, por ejemplo:

<i>echo</i>	<i>message</i>	<i>comment</i>	<i>results</i>
-------------	----------------	----------------	----------------

Con el parámetro **echo=TRUE** se establece que queremos mostrar el código fuente del chunk, este es el valor por defecto si no se especifica. En cambio, **echo=FALSE** indica que no queremos mostrar el código fuente.

(vgr. *echo=TRUE*) Para una variable aleatoria $X \sim \exp(\lambda)$ siempre se cumple que $F(E[X]) = 1 - e^{-1}$.

```
1-exp(-1)
```

```
## [1] 0.6321206
```

(vgr. *echo=FALSE*) Para una variable aleatoria $X \sim \exp(\lambda)$ siempre se cumple que $F(E[X]) = 1 - e^{-1}$.

```
## [1] 0.6321206
```

Con el parámetro **message=TRUE** establecemos que queremos mostrar los mensajes que R produce al ejecutar el código. En caso de **message=FALSE**, no se mostrarán.

(vgr message=TRUE) Cargamos el paquete de cuadrados mágicos

```
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
```

(vgr message=FALSE) Cargamos el paquete de cuadrados mágicos

```
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
```

Con el parámetro **comment=NA** se establece que no queremos mostrar los “#” que aparecen en el chunk.

(vgr. *comment=NA*) Para una variable aleatoria $X \sim \exp(\lambda)$ siempre se cumple que $F(E[X]) = 1 - e^{-1}$.

```
1-exp(-1)
```

```
[1] 0.6321206
```

Con el parámetro **results** se puede establecer la manera en la que queremos mostrar los resultados de la ejecución de un bloque de código. Este parámetro tiene cuatro posibles valores:

results=markup: Nos muestra los resultados en el documento final línea por línea, es el valor por defecto y coloca los “#”.

results=asis: Nos devuelve los resultados línea a línea de manera literal en el documento final y el programa con el que se abre el documento final los interpreta como texto y formatea adecuadamente.

results=hold: Muestra todos los resultados al final del bloque del código.

results=hide: No se muestra el resultado en el documento final. (parecido a la función `echo=FALSE`)

Existen cuatro valores para el parámetro `results`: `markup`, `asis`, `hold`, `hide`

El “`markup`” es el que por defecto viene, donde se muestran los “`#`”, a menos que se ponga `comment=NA`

El “`hold`” muestra todos los resultados juntos hasta el final del bloque

El “`hide`” oculta los resultados

El “`asis`” muestra los resultados de forma litera, interpreta el texto, en un cuadrado mágico devuelve cada fila una seguida de la otra.

En conclusión, al crear una chunk podemos considerar cuatro parámetros importantes: `echo`, `message`, `comment`, `results`