

Notas de Markdown

Carlos, C.

2022-07-31

Chunk

Los bloques de código de R en un documento Markdown se llaman **chunk** y son como el que se muestra a continuación:

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 hecho debemos realizar un **Knit**, esto se puede hacer dando click sobre el icono que aparece en RStudio o con el short cut *Ctrl+Shift+K*.

Ahí mismo podemos establecer si queremos que el Markdown sea mostrado como PDF o HTML.

Argumentos de un chunk

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

echo

message

comment

results

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. A diferencia de `echo=FALSE`, este último no muestra la línea de código de R pero sí los resultados. En caso de querer ocultar también los resultados podemos establecer `results=hide`.

Chunk en medio de texto

Es posible agregar un chunk dentro de un texto que estemos escribiendo sin la necesidad de hacerlo por separado (como hemos hecho hasta ahora). Para este fin se debe usar la sintaxis: `'r [código]'` en el espacio donde quisiéramos introducir el chunk.

(vgr) Para cualquier variable aleatoria $X \sim \exp(\lambda)$ se cumple que

$$F_X(E[X]) = 1 - e^{-1} = 0.6321206$$

Podríamos escribir dentro de un chunk “normal” el código de R necesario para realizar alguna tarea y luego establecer `echo=FALSE` y `results='hide'` para que no se muestre el código en el archivo PDF, pero hacer uso de los cálculos que R haya realizado con ese código que escribimos.

(vgr) Este año he hecho $n = 9$ exámenes, con una media $\bar{x} = 8.33$ y una desviación estándar de $s = 1.41$

El chunk que está implícito en la anterior línea es el siguiente:

```
notas=c(9,10,7,7,9,10,8,9,6)
media=mean(notas)
n=length(notas)
desv.tip=sd(notas)
```

Documentación de textos usando Markdown

Título 1

Título 2

Título 3

Título 4

Título 5 Esto es un texto llano. Podemoe escribir sin algún problema todo el texto que quisieramos para acompañar a tanto el código en R como las fórmulas en *LaTeX*.

Esto sería ua nueva línea de texto. Se puede poner este nuevo párrafo con una diagonal invertida (como en LaTeX común) o presionando dos “Enter” después de acabar un párrafo.

Para colocar cursiva podemos usar un **asterísco** *esto sería un texto en cursiva* o bien usar **barra baja** *esto también es cursiva*.

Para colocar negrita podemos usar **doble asterísco** o bien **doble barra baja**.

Los superíndices se pueden colocar con “^”: este curso vale por mi nota²

Para tachar una palabra usamos doble tilde “~”: ~~Matemáticas~~

Se puede asociar un link a un texto de la siguiente forma: [SOA website](#)

Un guión corto se hace con doble guión “-”: –

Un guión largo se hace con triple guión “-”: —

Una elipsis se puede hacer con tres puntos “.”: ...

En caso de querer incorporar una imagen: “” y dentro del paréntesis debemos especificar donde se encuentra la imagen. Se puede “subir nivel” (mover entre carpetas) usando “..\" :



Se puede colocar una barra divisora con tres asteriscos “***”:

Aquí se empieza otra cosa

Las citas se pueden colocar con un signo “>”:

Esto es una cita

También podemos crear una lista:

Listas no ordenadas:

- Item primero
 - Sub item 1.1
 - Sub item 1.2
 - Sub item 1.3
- Item segundo
- Item tercero

Listas ordenadas

1. Primer item
 - Sub item 1.1
 - Sub item 1.2
2. Segundo item
3. Tercer item
4. Cuarto item
5. Quinto item

Para colocar tablas:

Alumno	Nota	Edad
Juan Gabriel	3.5	30
Ricardo	5.6	55
María	7.3	20
Antonio	9.5	28

Este lenguaje se puede usar para el README de un Github

Existen pequeñas diferencias entre mostrar este Markdwon con PDF y HTML

Al momento de incluir gráficos en nuestro Markdwon pudiera ocurrir que la imagen es demasiado grande y quisieramos ajustarla o centrarla, para ello podemos definir (dentro de los parámetros del chunk) lo siguiente

```
fig.width=5      fig.height=5      fig.align="center"
```