

Sesión 1

Manuel Toral

29 de septiembre de 2018

Estructura general del curso

Este curso se inspira del trabajo diario y profesional con R; es decir, el curso no se contruye en mejorar dramáticamente sus habilidades en pensamiento computacional o en conocimiento estadístico, sino el uso de este lenguaje para hacer de su trabajo diario algo mucho más rápido, más estructurado y más eficiente.

Por tanto, la estructura del curso se encuentra presentada en la siguiente figura.

Los temas que cubriremos están descritos en el syllabus.

Estructura de la clase

Las sesiones durarán, efectivamente 3 horas. La hora restante es par aun pequeño brake de 30 minutos (o como convenga el grupo) y la última media hora será para resolver un ejercicio. Aquí quienes quieran quedarse a resolverlo juntos nos quedamos y quien se quiera o tenga que ir, puede hacerlo con la confianza de que no veremos material nuevo.

¿Qué es Markdown?

Markdown (*MD*) es un lenguaje de *marcado*, su terminación comúnmente es `.md`. Sin embargo, nosotros usaremos una versión “con sabor” (*flavored*) a R, con terminación `Rmd`. Esto significa que *MD* no sirve para programa o procesar datos. Simplemente nos permite estructurar, escribir y presentar. Este elnguaje es muy popular, incluso *WhatsApp* lo usa. Prueba escribir en tu *WhatsApp* lo siguiente: ````Te mando mi código, *es muy importante* e=mc^2````.

Como pueden ver, *MD* es tan útil porque permite identificar y enfatizar cuando las cosas son códigos o instruir a la computadora a cambiar el tipo de letra con base en el código. Sin clics de más, sin problemas de edición y sin depender de software como *MS Office*.

¿Por qué empezar por *MD* y no por R?

1. Porque uno de los grandes retos del aprendizaje profesional en R es saber *cómo usarlo* en un contexto profesional. Yo creo, es debatible supongo, que de nada sirve poder hacer redes neuronales convolucionales (cosa muy cool y posible en R), si no podemos comunicarlo y entenderlo en un contexto del día a día.

En *STATA* por ejemplo, no se pudo usar *MD* para crear documentos desde este software hasta su última versión. Además, si se meten tantito, verán que es un problema. La distancia entre hacer un análisis en *STATA* o en *SPSS* o en *SAS* y presentarlo es larga. Gracias a *MD*, ésta no sólo se reduce si no que hace documentos que no sólo por ser técnicos se vean como salidos de una impresora de matriz.

2. Porque permite estructurar nuestro pensamiento para resolver problemas. Para construir efectivamente un documento en *MD* necesitamos conocer con antelación la estructura. Esto nos ahorrará mucho tiempo y nos permitirá partir todos los problemas que tenemos en pequeñas partes. Escribir un documento, por

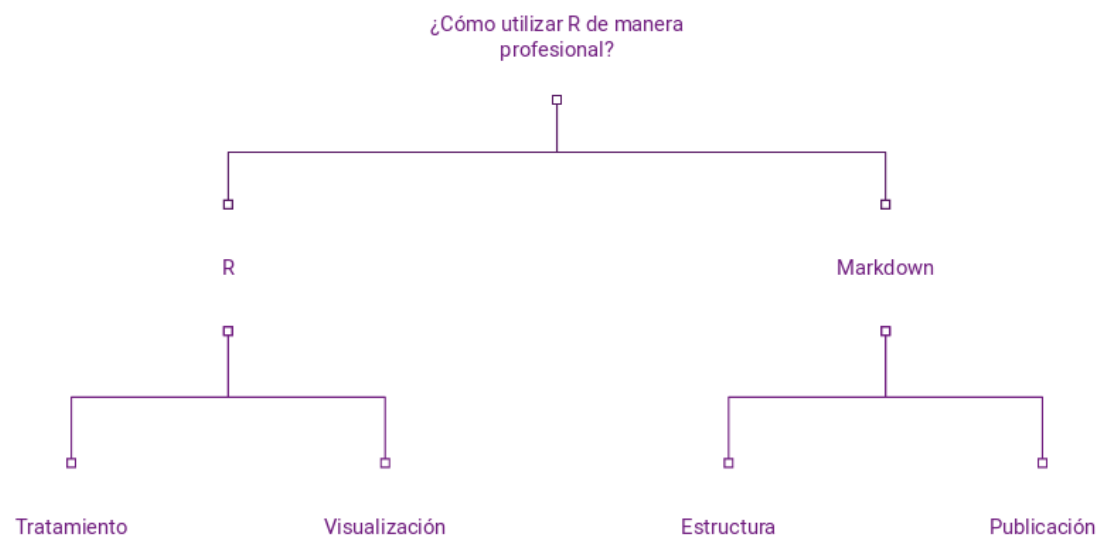


Figura 1: Esta va a ser la estructura general del curso

ejemplo, es un problema. Piensen en su tesis o en algún reporte. Escribir con una estructura permite que ese gran problema se resuelva en pequeñas ideas.

¿Cómo usar *MD*?

Para estas alturas, ya pudieron haberse dado cuenta de que un archivo *Rmd* es, finalmente, un archivo de texto con algunas instrucciones que le indican a nuestra computadora cómo construir un documento, pero vez de usar clics de nuestro ratón usamos instrucciones escritas. De esta forma, que funcione o no nuestro documento está en función del código y no de los clics que demos. Así, un documento *MD* en el sabor de *R* se compone de 3 elementos:

1. Un *YAML*: Antes que **cualquier** cosa, es necesario crear un “encabezado” que indique qué queremos que sea nuestro documento. Un *YAML* correcto debe indicar el título, la persona autora, la fecha y qué se espera que sea el resultado. *MD* puede producir documentos en *HTML*, *PDF* (vía *L^AT_EX*) y *MS WORD* (para los que les guste la mala vida).

```
---
title: "El mejor reporte de la historia"
author: "Manuel Toral"
date: "Septiembre 29, 2018"
output: html_document
---
```

2. Una estructura. Las estructuras en *MD* se componen por el “signo de gato” *#*. Un *#* indica la primera estructura, *##* indica una estructura “hija” o dependiente, un *###* indica una estructura dependiente de la anterior... y así. Por ejemplo:

```
# Así se ve un título

## Así se ve un subtítulo
Este es el contenido de esta sección.

### Así se ve un subsubtítulo
Este es otro contenido de esta sección.
```

Así se ve un título

Así se ve un subtítulo

Este es el contenido de esta sección.

Así se ve un subsubtítulo

Este es otro contenido de esta sección.

3. Una serie de instrucciones contenidas en **chunks** de código. un **chunk** es un espacio en el que *MD* reconoce que lo que se encuentre dentro va a ser una instrucción en algún lenguaje de programación. Todo lo que se encuentre dentro va a ser considerado parte de ese lenguaje y correrá en nuestra consola como instrucciones. **Esta característica es la más poderosa y útil de MD.** Para abrir un **chunk** hay que abrir y cerrar. Nunca dejar abierto, nunca olvidar alguna de estas partes. Así, para escribir un pedazo de código debemos encerrarlo en medio de estas dos llaves: “`{r}`” y “”

Así se ve un subsubtítulo

Este es otro contenido de esta sección. “