R para Ciencia de Datos: Introducción (Capítulo 1)

Gabriel Illanes

17/3/2021

¿Qué vamos a estudiar?

El objetivo de la ciencia de datos es transformar datos crudos en conocimiento sobre un área o problema. El objetivo del libro "*R* para Ciencia de Datos" de Wickham y Grolemund (y también de nuestro seminario) es comprender como el software *R* nos puede ayudar en la ciencia de datos. Si quisieramos esquematizar cómo es el desarrollo de un proyecto de ciencia de datos, se podría ver de la siguiente manera:

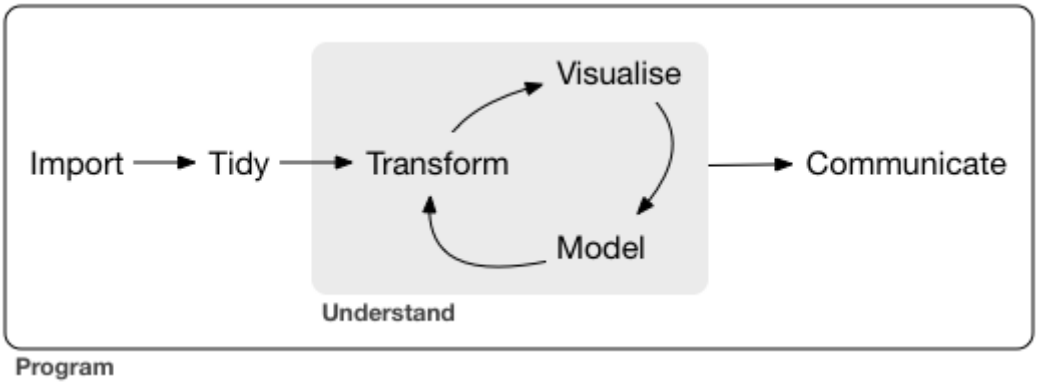

Figure 1: Diagrama de un proyecto de ciencia de datos. Imagen obtenida del libro "R para Ciencia de Datos"

Desglocemos los aspectos involucrados en el diagrama:

- **Importar los datos**: Necesitamos poder cargar la base de datos de nuestro interés en *R* para poder trabajar en ellos. La base de datos puede estar en distintos formatos, y *R* provee recursos para poder importarlos para los formatos más usuales.
- **Limpiar los datos**: Los datos crudos (recién importados) no suelen estar en un formato conveniente para su análisis. Es importante que los datos queden cargados en *R* en un formato de *data frame*: tenemos observaciones (filas del *data frame*) de fenómenos, a los cuales se les mide cantidades variables (columnas del *data frame*). En lo posible, cada variable debe tener un nombre (para acceso más conveniente), y los tipos de objetos en cada columna tiene que ser siempre el mismo (cada columna con su tipo).

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

- **Transformar los datos**: La transformación implica reducir las observaciones a aquellas que sean de interés (como todas las personas de una ciudad o todos los datos del último año), crear nuevas variables que sean funciones de variables ya existentes (como calcular la rapidez a partir de la velocidad y el tiempo) y calcular una serie de estadísticos de resumen (como recuentos y medias).
- **Visualización de los datos**: La visualización es sumamente importante, sobre todo en un análisis exploratorio de los datos, ya que podemos observar patrones a ojo, que sería difícil encontrarlos en los datos con algoritmos, porque muchas veces no sabemos donde empezar a buscar. El problema es que este proceso no puede ser automatizado, ya que requiere la interpretación de una persona.
- **Modelado de los datos**: Una vez que generamos buenas preguntas, los modelos proveen herramientas para dar respuestas concretas, basándonos en ciertos supuestos. Sin embargo, si un modelo falla, puede ser difícil saber la razón. Para este aspecto es importante un vasto conocimiento en matemática y estadística.
- **Comunicación**: Es vital para el trabajo de un analista de datos poder comunicar de manera eficiente los resultados obtenidos. Esta comunicación puede darse tanto al final del análisis (comunicación de resultados finales) como durante el análisis (discusión de resultados intermedios, para decidir el avance del proyecto).
- **Programación**: Todos los ítem anteriores están influenciados por la programación. Hoy por hoy, la programación es una parte fundamental de cualquier análisis de datos. Uno no necesita ser un experto programador, pero un manejo fluido del lenguaje de programación en cuestión ayuda a lograr mejores resultados y eficiencia.

En el libro se enseñarán herramientas para afrontar cada uno de estos aspectos, pero por motivos pedagógicos, no necesariamente en el orden antes presentado. La idea es, aparte de aprender análisis de datos, mostrar que es un trabajo interesante y entretenido!

Que cosas *no* vamos a estudiar:

Es importante para nosotros mantener claro el alcance del seminario y del libro:

- **Big Data**: Lo que usualmente se conoce como *big data* consiste en la ciencia de datos, cuando los datos resultan muy grandes para ser manejados con herramientas clásicas de matemática, o herramientas usuales de programación (sea hardware o software). Muchas de las complicaciones que surgen son técnicas y complejas, y no se entienden del todo sin un conocimeinto sólido de escenarios más sencillos. El software *R* provee herramientas para *big data*, pero no está en el alcance del libro (ni del seminario).
- **Otros lenguajes de programación**: *R* no es la única opción para el análisis de datos; también contamos con otros lenguajes potentes, amigables y de software libre. Dos claros ejemplos son *Python* y *Julia*, pero también es común manejar lenguajes de más bajo nivel, como *C++* o *Fortran*. Es importante saber que todo lenguaje tiene sus fortalezas y debilidades, por lo que conviene poder manejar varios lenguajes complementarios, pero también es muy importante tener un muy buen manejo de, al menos, un lenguaje (probablemente eso es mejor que tener un manejo razonable de muchos lenguajes). Este libro se dedica al estudio del software *R* únicamente, y apunta a que el estudiante alcance un nivel avanzado para poder afrontar distintos escenarios en un proyectos de ciencia de datos.
- **Datos no rectangulares**: Son aquellos que no se adaptan al esquema de observaciones y variables discutido anteriormente. Ejemplos de datos no rectangulares son imágenes, sonidos, árboles y texto. Sin embargo, los data frames son una estructura muy flexible y usada en muchísimos escenarios.
- **Confirmación de hipótesis**: En ciencia de datos se dan dos escenarios. El primero es cuando tenemos una base de datos, y varias preguntas específicas que queremos responder, a eso llamamos *confirmación de hipótesis*. En el escenario de confirmación de hipótesis, necesitamos un modelo matemático específico que nos ayude a responder cada pregunta, y las respuestas de esos modelos son finales (no se pueden reutilizar los datos, para evitar fenómenos como el *p-hacking*). El segundo escenario es cuando disponemos de una base de datos, pero no hay formuladas preguntas todavía que queramos responder. Simplemente, disponemos de datos interesantes, relevantes en un área, y queremos ver qué nos pueden decir; a eso se

lo llama *análisis exploratorio* o *generación de hipótesis*. Es importante entender que el seminario apunta principalmente a la parte computacional de la ciencia de datos, y no en la parte matemática estadística del modelado (que suele ser particularmente compleja en un escenario de confirmación de hipótesis).

Para comenzar a trabajar:

Para poder trabajar en este seminario, se necesita tener disponible las siguientes herramientas

- **R**: Pueden descargar el software *R* de la página https://cloud.r-project.org.
- **RStudio**: Es el ambiente de desarrollo integrado (IDE) estándar para el uso de *R* hoy en día. Es una herramienta de software libre muy potente, que provee muchas herramientas útiles para una buena experiencia programando en *R*. Luego de descargar *R* e instalarlo, pueden descargar *RStudio* de la página http://www.rstudio.com/download.
- **tydiverse**: Es un paquete del software *R* que contiene una gran cantidad de herramientas complementarias para la ciencia de datos. Este paquete apunta a un estilo de programación que es muy amigable: permite realizar código claro, eficiente y ordenado para todas las instancias de la ciencia de datos. Una vez que tengan instalado *R* y *RStudio*, pueden instalar *tidyverse* ejecutando la línea de código

```
install.packages("tidyverse")
```

- **datos**: Es un paquete que contiene distintos data frames, utilizados en distintos ejemplos del libro. Para poder ejecutar el código de ejemplo del libro traducido al español„ lo van a necesitar (si van a basarse en la versión original del libro, no necesitan instalar nada). Como cualquier otro paquete, instalan el paquete *datos* ejecutando la línea de código

```
install.packages("datos")
```

¿Qué hacer si necesitan ayuda?

Lo más razonable es que, mientras preparan una presentación en el seminario (y más aún trabajando en cualquier proyecto de ciencia de datos), van a surgir preguntas, dudas, inquietudes, errores, y más. Algunos piques sobre distintos recursos con los que cuentan:

- **Consultar a los docentes**: a los responsables de este seminario nos parece lo más normal del mundo que necesiten hacernos preguntas durante la preparación de sus presentaciones, así que no duden en consultar si lo necesitan.
- **Discutir con sus compañeros**: pueden discutir con sus compañeros conceptos e ideas generales sobre los temas que tengan que preparar, e incluso ideas generales sobre código. Sin embargo, la redacción de la presentación y el código debe corresponder únicamente al que presenta.
- **Buscar en Google**: si tienen una duda, o quieren lidiar con un error, lo más probable es que alguien más ya haya pasado por lo mismo, y eso haya generado alguna entrada en Internet que ustedes puedan aprovechar. Recuerden hacer buenas búsquedas en Google (es todo un arte que hay que entrenar).
- **Pedir ayuda en foros**: algunos foros en internet se especializan en el intercambio sobre temas de ciencia o programación entre distintos usuarios y expertos. El ejemplo más conocido es *Stack Overflow*. Es muy posible que muchas de sus búsquedas en Google terminen en una entrada de Stack Overflow, e incluso cada uno puede plantear una consulta (en dicho caso, ver en el libro los piques para hacer buenas consultas, es muy importante). Recuerden también usar el *foro de la la página del seminario* en EVA.

Más sobre el seminario

Este documento pretende ser un ejemplo de cómo podría verse la presentación del capítulo 1 del libro, que corresponde a la introducción. La idea es seguir fuertemente el libro, pero también mostrar que ustedes

procesaron y asimilaron el contenido, y pueden parafrasear, agregar o quitar contenido para mejorar la presentación (copiar y pegar del libro a la presentación no va a estar muy bien visto).

Como se dijo antes, si bien se puede discutir con los compañeros u otra gente sobre ideas generales para la presentación, o ideas generales sobre código (pseudocódigo), tiene que estar claro que la redacción de la presentación, incluido su código, es individual.

Es importante que generen un documento como este y lo compartan antes de presentar, para que sus compañeros dispongan de él durante la presentación. Hay muchas variantes sobre el tipo de documento que pueden generar. Puede ser modo artículo (como este) o diapositiva; pero en cualquier caso, van a necesitar cierto dominio de *R Markdown* (no está para nada recomendado el uso de PowerPoint, Word u otras herramientas similares). El código que se usó para generar este documento estará disponible para ustedes, y a partir de él debería ser fácil aprender las básicas.

Se espera que, durante la preparación de las presentaciones, los estudiantes hagan los ejercicios que se encuentran en los capítulos. No hace falta que los escriban o los presenten durante el seminario, pero durante las presentaciones, los docentes les podemos hacer preguntas sobre algunos de los ejercicios.

Además, es importante que sepan que se espera que todos los estudiantes lean todos los capítulos antes de la presentación correspondiente, y que aporten dudas y comentarios durante la presentación para así enriquecerla.