

Sí pero no, *¿Qué hacer cuando nada es suficiente?*

Palabras clave: paquetes, alternativas, flowchart

Abstract

R cuenta con un vasto universo de paquetes, diseñados para abarcar un abanico de tareas demasiado amplio para ser descrito. No obstante, podemos encontrarnos fácilmente en una situación donde no exista paquete alguno que solucione por sí solo nuestras necesidades. Esto plantea la pregunta “¿Qué hacer cuando ningún paquete es el idóneo para mí tarea?”

El problema

En muchas ocasiones de nuestra vida laboral, institucional o incluso cotidiana nos encontraremos con tareas que pueden y/o deben abordarse utilizando R. Gracias a que R es parte de un proyecto colaborativo y abierto, cuenta con un número tan amplio de paquetes desarrollados por usuarios que, casi con seguridad, existirá uno o más de ellos que puedan ser de utilidad para dicha tarea.

Pero, ¿qué ocurre si ninguno de los paquetes disponibles satisface al 100% las necesidades específicas de la tarea a resolver? Un ejemplo concreto es el disparador de este trabajo: tener que crear un flowchart o diagrama de flujo para un ensayo clínico con ciertas características puntuales requeridas por el Centro Coordinador de Datos del proyecto que ninguno de los paquetes explorados podía cumplir en su totalidad.

La solución

Varios caminos se abren ante el dilema planteado. Si bien en algunos pocos casos las características que no se puedan cumplir con ningún paquete podrán ser eventualmente descartadas de no ser sumamente necesarias, en general esto no será posible. Entonces, nuevas alternativas deberán ser buscadas para poder cumplir con todos los requerimientos solicitados.

Una elección, para nada recomendada pero existente, es la modificación manual de los resultados que arroje R para poder salvar las diferencias entre lo que se puede hacer con el software y lo que se pide como requerimiento. Esta solución no solo atenta contra el concepto de reproducibilidad, sino que también incrementa enormemente la posibilidad de errores al mismo tiempo que dificulta que la tarea en cuestión sea llevada a cabo periódicamente de forma sencilla.

No muy lejos de esta solución, se encuentra la opción de complementar con otro software los resultados que arroje el primero. Esto, si bien disminuye la posibilidad de error y simplifica ligeramente la tarea en caso de tener que ser repetida, no deja de ser poco práctico

SI PERO NO, ¿QUÉ HACER CUANDO NADA ES SUFICIENTE?

y contraproducente para que alguien más replique los resultados, ya que necesitará más conocimiento y herramientas para lograrlo.

Tomando la idea de complementar, pero sin salirse de R, surge la alternativa de utilizar más de un paquete para poder franquear las limitaciones a las que uno se vería enfrentado en caso de utilizar solo uno de ellos. No debe confundirse este enfoque con el hecho de utilizar varios paquetes a lo largo de un script, todos con funcionalidades diferentes. En este caso, se hace referencia a recurrir a paquetes diseñados para el mismo propósito, combinando una o más de sus funciones, con el fin de alcanzar el resultado deseado. Esta idea, si bien más práctica que las anteriores, en muchos casos colisiona con la dificultad de mezclar funciones de diferentes paquetes debido a cómo estos están contruidos.

Otra solución, mucho más ambiciosa que las presentadas hasta ahora, sería la de “empezar desde cero” y solucionar las necesidades que trae aparejada la tarea en cuestión mediante la creación de un nuevo paquete que permita cumplir los requerimientos del usuario en su totalidad. Naturalmente, esta alternativa puede llevar mucho tiempo de trabajo por lo que no sería muy viable en la práctica, sobre todo en situaciones donde existen fechas límites a respetar.

La última de las soluciones propuestas es, quizás, la menos utilizada. Consiste en tomar un paquete que satisfaga algunas de las necesidades específicas de la tarea a resolver y modificar sus funciones o crear algunas nuevas que complementen lo ya existente para poder cumplir con todos los requerimientos que se deseen. Esta alternativa conlleva que el usuario primero debe interiorizarse con el funcionamiento de las funciones del paquete para poder luego realizar modificaciones a conciencia o desarrollar nuevas funciones que se puedan integrar al paquete y a su lógica de funcionamiento interna.

De todas las soluciones que fueron mencionadas, esta última fue la que se utilizó para poder obtener el flowchart que cumpliera con los requerimientos pedidos. Utilizando como base el paquete “Gmisc”, se tomaron algunas de sus funciones y se adaptaron para incluir argumentos en las mismas que permitieran obtener resultados ligeramente diferentes pero que cubrieran las necesidades de la tarea. Luego de modificar las funciones necesarias, se las agregó al environment del paquete para que puedan utilizar otras funciones internas del mismo. De esta forma, la función modificada terminó siendo, a la vista de R, una más del paquete original.