R en producción: aprendizajes, retos y mejores prácticas

Ángel Escalante , Nancy Morales

Abstract Con más de tres años de experiencia trabajando como desarrolladores en R, compartimos las experiencias, retos y aprendizajes a los que nos hemos enfrentado en la industria al usar R como un medio para el procesamiento de datos, análisis y generación de reportes. Asimismo, proponemos un conjunto de puntos, a los que denominamos *imprescindibles*, tales como: apegarse a un esquema de trabajo de desarrollo (p. ej. SCRUM), elegir una versión de software adecuada a las necesidades como Microsoft R Open o R, emplear librerías que optimicen el uso de memoria y permitan crear paquetes escalables e implementar testing (unitario y de regresión) riguroso para mayor confiabilidad. Estos puntos nos han ayudado a mantener modelos estadísticos y comunicar resultados orientados a clientes en un esquema automatizado y confiable. De igual forma, dicho esquema nos ha permitido tener tiempos de respuesta rápidos ante bugs en producción y así reducir el impacto en costos que estos puedan tener.

Palabras clave: R - producción - paquetes - metodología

Introducción

Las organizaciones hoy en día necesitan sistemas que ejecuten continuamente código para apoyar la toma de decisiones de los clientes tanto internos como externos. Esto implica asegurar un funcionamiento correcto y eficiente de dicho sistema para cumplir con la demanda de los usuarios. Por lo tanto, características como contar con código legible, estable y escalable, uso de recursos y tiempo de ejecución (entre otros) son fundamentales en ambientes de producción. Estas características se traducen en evitar costos por bugs inesperados y generar satisfacción para los usuarios. Las compañías han volteado a ver las herramientas Open Source como claves para su desarrollo. Hoy en día, grandes compañías transnacionales como Netflix, AT&T, PayPal, entre otras, consideran software open source como pieza fundamenta en su funcionamiento. R es un lenguaje de código abierto para estadística, visualizaciones y machine learning. Con R puedes generar desde una tabla resumen de datos hasta un dashboard que permita a los usuarios/clientes revisar y analizar su información en tiempo real. Sin embargo, al no ser un lenguaje multipropósito, ¿Es R un lenguaje que puede ser usado en Producción?

Después de haber trabajado en el sector privado y dar mantenimiento a paquetes en R que ejecutan cerca de 3,500 análisis mensuales dentro de la organización para entrega de reportes, modelos e incluso dashboards de análisis, proporcionaremos nuestras experiencias, retos y aprendizajes sobre nuestro uso de R en una organización de alta demanda.

Flujo de trabajo como desarrollador

Existen metodologías de trabajo en desarrollo de software denominadas *"metodologías Ágiles"*, las cuáles son esenciales conocer para trabajar en un equipo de desarrollo y, sobre todo, a distancia. Trabajamos bajo la metodología ágil SCRUM, la cual está diseñada para proyectos en entornos de alta demanda, de inmediatez, que busca la innovación y la productividad. Esto implica, organización y coordinación entre los desarrolladores con la persona que tiene la responsabilidad de definir los alcances y expectativas de nuevos desarrollos, actualizar desarollos previos y definir proyectos de innovación. En el flujo de trabajo, se establecen olas de trabajo (mejor conocidos como sprints) que son periodos de 3 semanas, cada una con sus etapas de desarrollo: definición de features y/o bugs del sprint, desarrollo (escribir código), testing, presentación de resultados y retrospección del sprint.

Lo imprescindible para R en producción

En nuestra experiencia, hemos encontrado puntos que encontramos imprescindibles para ser un buen desarrollador en R en un ambiente de producción:

- **Adoptar una metodología de trabajo ágil**. Apoyarse de herramientas y paquetes que te permitan seguir el flujo de la metodología lo más fácil posible. Esto implica, auxiliarnos de herramientas de código colaborativo y control de versiones, Git, Azure DevOps, Amazon Web Services, Google Cloud o GitHub.
- **Establecer una versión de R** en la cuál correrán los análisis, aplicaciones o producto el final que se quiera entregar. Esto hará que los paquetes funcionen de forma esperada y reducirá el riesgo de enfrentar bugs por actualización de versiones.

- **Uso de paquetes de desarrollo** especializados como **`devtools`**, **`usethis`** y **`testthat`**. Así como seguir las mejores prácticas en desarrollo de paquetes propuestos por Jenny Bryan y Hadley Wickham en su libro R Packages (Hadley, 2015)[1].
- **Código legible, claro y documentado**. A veces, es mejor tener más líneas de código entendibles, que un código de pocas líneas, pero críptico (Dustin & Trevor, 2011)[2].
- **Crear un ecosistema de paquetes** intercomunicados y con propósitos específicos. Esto ayudará a simplificar todo el flujo del análisis que se quiere entregar. Si el propósito de la solución involucra recibir, análizar y presentar datos, tener un paquete dedicado a cada uno de estos pasos es lo deseable.
- **Memoria y uso de recursos**. Los recursos siempre son limitados, escribir código que sea eficiente en términos de memoria es fundamental. Revisaremos algunos paquetes, como **`data.table`**, que pueden ser de ayuda cuando se manejan volúmenes de datos grandes.
- **Regression testing**. Estamos trabajando con un paquete que entrega un reporte, después de un mes de desarrollo, ¿este reporte no ha sido involuntariamente alterado? ¿el nuevo código interfiere el anterior? Además del unit test, el cual es obligatorio si queremos evitarnos problemas de malfuncionamiento del paquete, el propósito del regression test toma el resultado de un producto final que definido como aceptable y comparamos con el resultado después de haber hecho diversos cambios. Esto asegurará que el código sigue produciendo los resultados esperados.

Conclusiones

Los ambientes de producción no son exclusivo de lenguajes multi-propósito como Python, JavaScript, etc. R tiene todo lo necesario para poder ser usado como lenguaje backend en aplicaciones web, herramienta de análisis principal y de reporteo en ciencia de datos entre muchas cosas más. Quienes venimos de escuelas de matemáticas quizá estemos más familiarizados con R que con otros lenguajes comunes en la industria de software y conceptos técnicos del desarrollo. Sin embargo, es la comunidad de R y los múltiples recursos de fácil acceso son los que permiten adentrarse en el mundo de DevOps e iniciar un camino como desarrollador en R en específico. El camino no es sencillo y habrá que aprender bajo prueba y error, pero con el tiempo entender estas capacidades, adoptar una metodología y aprender de otros desarrolladores, permitirá ayudar a la organización sin la necesidad de salir de R.

Referencias

[1] Wickham, Hadley. 2015. *"R Packages"* . USA: O'Reilly

[2] Boswell, Dustin, and Foucher, Trevor. 2011. *"The Art of Readable Code: Simple and Practical Techniques for Writing Better Code."* USA: O'Reilly