

imarpe: un paquete para la automatización de gráficos, tablas y reportes usando R

Criscely Luján-Paredes

24 de Julio de 2017

Resumen

Este documento es una introducción al uso del paquete **imarpe**, el cuál proporciona herramientas para la elaboración de gráficos, tablas y reportes que se realizan de manera rutinaria en las investigaciones producidas por el Instituto del Mar del Perú (IMARPE). El objetivo principal de **imarpe** es que los usuarios trabajen en R sin requerir conocimientos avanzados de programación, ahorrando tiempo en el procesamiento de información.

Palabras clave: R, imarpe, automatización, reportes, figuras, tablas.

Introducción

imarpe es un paquete implementado en R cuya principal función es automatizar el procesamiento de información así como automatizar la elaboración de gráficos, tablas, y reportes que son necesarios en el trabajo rutinario para el personal del Instituto del Mar del Perú (IMARPE). El diseño de este paquete es flexible, permitiendo al usuario realizar cambios sobre los parámetros de los resultados ya tendrán a su disposición el código en R que genera los productos; adicionalmente, el diseño de este paquete permite trabajar con funciones genéricas (e.g. plot, summary, print) sobre cada clase de datos, facilitando de esta manera su utilización incluso con conocimientos mínimos de R.

Dentro de las principales ventajas que proporciona **imarpe** están:

- No requiere que el usuario posea conocimientos avanzados de programación y/o R,
- Permite ahorrar tiempo en el procesamiento de la información,
- Brinda al usuario un procesamiento automatizado de la información,
- Permite que el usuario realice cambios sobre los parámetros de las salidas (tablas, figuras y reportes.)
- Proporciona gráficos de buena calidad así como la capacidad de realizar análisis reproducibles.

Programación orientada a objetos

R tiene tres sistemas orientados a objetos (S3, S4 y R5), y todos trabajan con los conceptos “clase” y “método”. Una clase se define como un tipo de objeto, describiendo que propiedades posee, como funciona y como este objeto se puede relacionar con otros tipos de objetos; de esta manera, cada objeto debe poseer una clase. Por otro lado, un método se define como aquella función que está asociada a un tipo particular de objetos.

S3 implementa un estilo de programación orientada a objetos llamada función genérica OO (o en inglés, generic-function OO). En el funcionamiento de este sistema se realizan cálculos a través de métodos y a través de un tipo especial de función llamada “función genérica”, la cual decide qué método llamar.

El uso principal de S3 en R es a través de los métodos print, summary and plot. Estos métodos permiten tener una función genérica por ejemplo de print, que mostraría el objeto de una manera especial.

Instalación de imarpe

Para usar **imarpe** se debe tener instalado R, y se recomienda usar la interface gráfica de R Studio. Para la instalación de todo paquete de R se requiere contar con internet y luego de verificar esto, desde R Studio se debe intalar y cargar el paquete devtools, de la siguiente manera:

```
install.packages("devtools")  
library("devtools")
```

La instalación de **imarpe** se realizará directamente de la cuenta github del Instituto del Mar del Peru (IMARPE) <https://github.com/imarpe>, lugar donde se trabaja en las mejoras y actualizaciones de los paquetes en R desarrollados por el personal del IMARPE.

En esta web también se encuentra **imarpe**, y su instalación se realizará usando las siguientes líneas de código. La primera línea de código instala el paquete y debe ser corrida por única vez, sin embargo, cada vez que se haga una actualización del paquete, se deberá correr nuevamente para cargar de manera automática la nueva versión de **imarpe**. La segunda línea de código se encargará de cargar el paquete la instalado, esta debe ser ejecutada cada vez que se quiera usar el paquete.

```
install_github("imarpe/imarpe")  
library(imarpe)
```

Estructura de imarpe

imarpe cuenta con el desarrollo del módulo de pesquería. Sin embargo, el objetivo es extender el desarrollo del paquete a tres módulos más, los cuales estén conformados por los modulos de biología, oceanografía y cruceros hidroacústicos.

Módulo Pesquería:

Esta sección trabaja sobre información pesquera, orientada al análisis de las variables: captura y desembarque, esfuerzo y captura por unidad de esfuerzo (cpue), reproduciendo tablas, gráficas y reportes de manera automatizada. Así mismo, el Programa de Observadores a Bordo (Programa de Bitácoras de Pesca) del IMARPE también cuenta un subsección de trabajo dentro del módulo de pesquería, el cual permite obtener las principales gráficas y tablas que son requeridas en los reportes periódicos que emiten.

Las principales funciones con las que cuenta **imarpe** son 4 y las abordaremos una a una mediante sub-secciones que describan las clases de objetos.

```
getFishingData  
getBitacoraData  
getMainResults.bitacora  
getDailyReport
```

1. Get the fishing data:

Esta función principal es usada para dos tipos de clases de datos: fishery y cpue. La primera clase, **fishery**, contiene variables pesqueras que en este caso incluye a los desembarques (landing) y al esfuerzo (effort). La segunda clase de datos, **cpue**, incluye a la captura por unidad de esfuerzo (cpue).

imarpe incluye una base de datos (**fisheryData**) para ejemplificar el uso de esta función. Sin embargo, toda base de datos debe poseer información sobre el tiempo (año, mes y día), sobre la especie, sobre el tipo de flota y puerto, la información de posición (que es opcional), e imprescindible la información de la captura (o

desembarque), así como las variables que serán utilizadas en el esfuerzo pesquero. Se puede utilizar la ayuda de `fisheryData` para conocer un poco más de su contenido.

El nombre de las columnas debe ser el mismo al de la base de datos de ejemplo, sin embargo el uso de mayúsculas y minúsculas será resuelto internamente.

Luego se utilizará la `getFishingData` para analizar los desembarques de la base de datos bringada. La ayuda de esta función puede ser visualizada para una mejor comprensión de sus parámetros.

```
# Cargar la base de datos
?fisheryData
data(fisheryData)

# Usar getFishingData con la base de datos ejemplo (fisheryData)
?getFishingData
desembarque = getFishingData(file = fisheryData, type = "fisheryInfo", varType = "landing",
                             sp = "caballa")
```

`getFishingData` lee información pesquera para obtener i) un `data.frame` (o marco de datos) y ii) una lista con las principales características de dicho `data.frame`.