

Reporte de Seguimiento - RepSeg

AFRTAM

19 de septiembre de 2016

Este manual muestra los pasos para la instalación y aplicación del paquete **RepSeg** usando RStudio.

Instalación del paquete **RepSeg**

- 1) Escribir las siguientes líneas para instalar los paquetes **devtools**, **Hmisc** y **R.utils**:

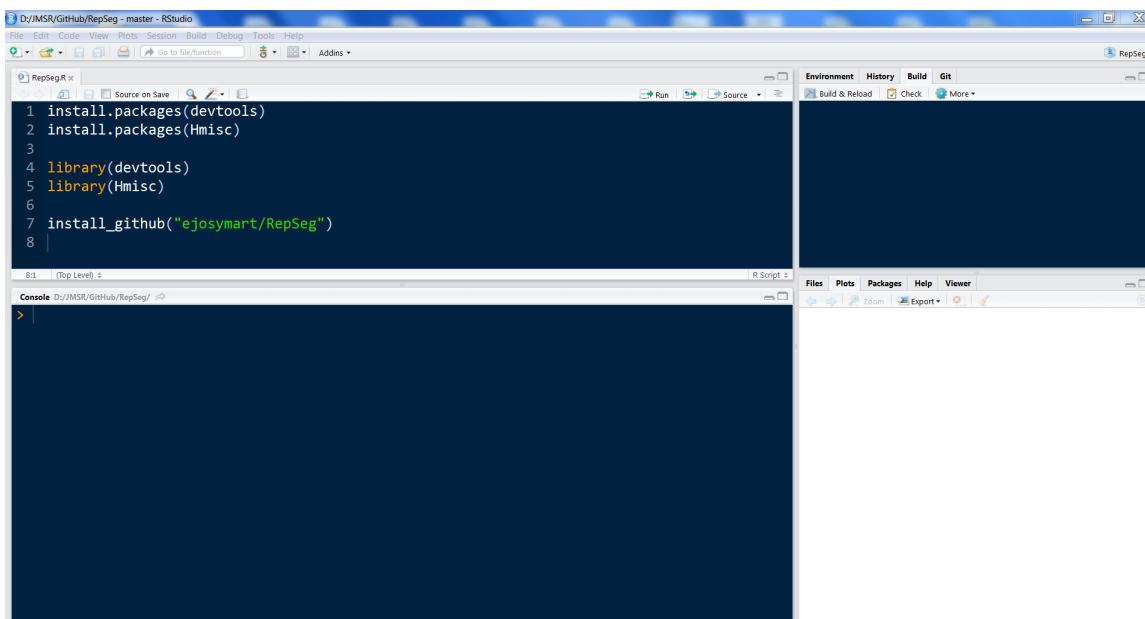
```
install.packages(devtools)
install.packages(Hmisc)
install.packages(R.utils)
```

Estos paquetes se instalan solo una vez.

- 2) Para instalar el paquete **RepSeg** copiar las siguientes líneas:

```
library(devtools)
library(Hmisc)
library(R.utils)
install_github("ejosymart/RepSeg")
library(RepSeg)
```

Cargar los paquetes usando el comando **library**. La última línea de código se debe correr solo una vez. Sin embargo, cada vez que se haga una mejora o actualización al paquete **RepSeg**, se deberá correr esta línea nuevamente. Para correr cualquier línea de código, se deberán ubicar en la línea y ejecutarla usando las teclas “Ctrl” y “R” al mismo tiempo.



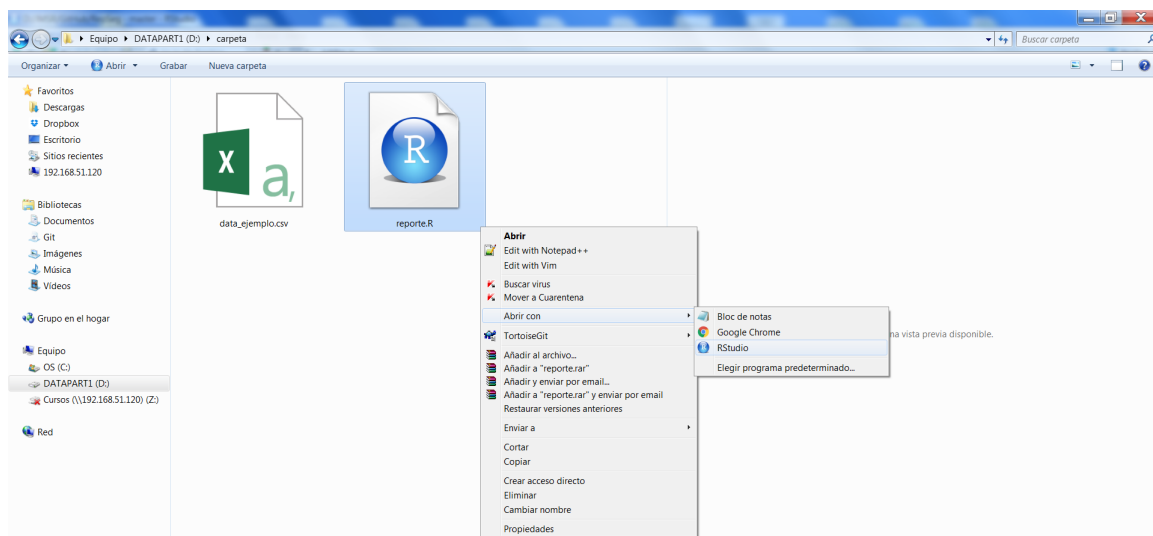
Aplicación del paquete *RepSeg*

Una vez instalado el paquete *RepSeg*, para poder obtener el reporte se necesita tener una base de datos en extensión *.csv*. Esta base de datos debe tener el siguiente formato:

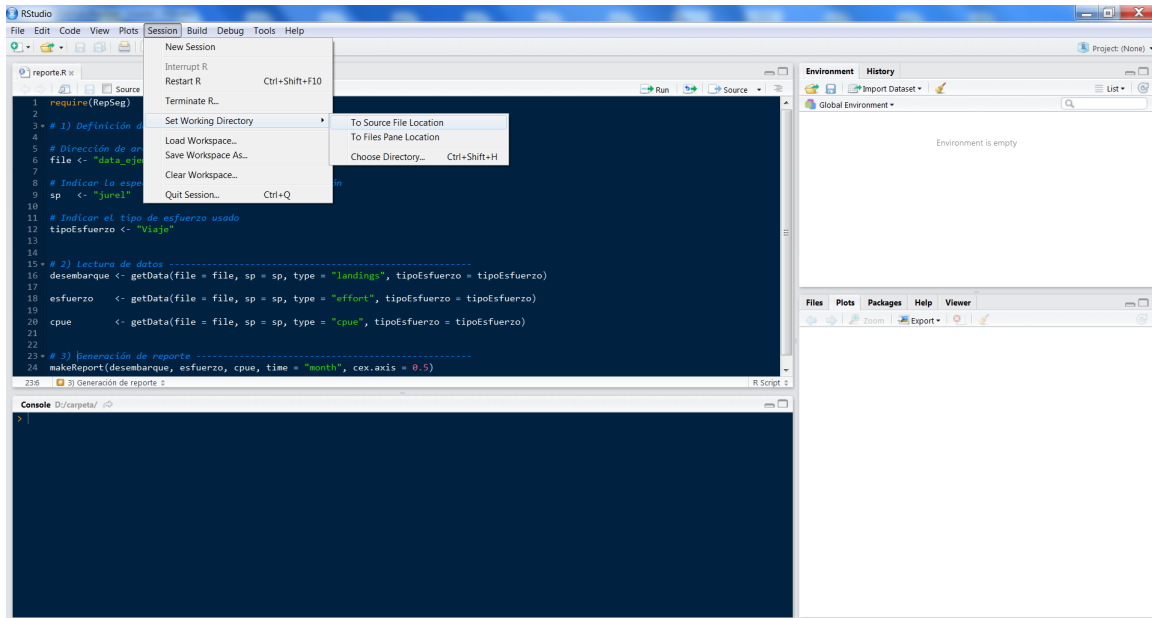
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	year	month	day	pto pizzaro_c	pto pizzaro_e	acapulco_c	acapulco_e	cancas_c	cancas_e	la cruz_c	la cruz_e	zorritos_c	zorritos_e	mancora_c	mancora_e	cabo l
2	2016	jan		1				115	10			121	4			
3	2016	jan		2								138	3			
4	2016	jan		3		56	5					221	6			
5	2016	jan		4		336	20					128	7			
6	2016	jan		5								371	11			
7	2016	jan		6		72	11					99	3			
8	2016	jan		7												
9	2016	jan		8												
10	2016	jan		9												
11	2016	jan		10												
12	2016	jan		11												
13	2016	jan		12		16	2					194	5			
14	2016	jan		13								91	4			
15	2016	jan		14												
16	2016	jan		15								226	6			
17	2016	jan		16										170	3	
18	2016	jan		17								57	1			
19	2016	jan		18								80	2			
20	2016	jan		19				6	1			15	1			
21	2016	jan		20												
22	2016	jan		21												
23	2016	jan		22								97		105	2	
24	2016	jan		23		20	1									
25	2016	jan		24				19	1			165	2	160	2	
26	2016	jan		25								33	1			

Las primeras tres columnas indican, el año, mes y día respectivamente. Las siguientes columnas corresponden a los datos de captura y esfuerzo por puerto. Nótese que los nombres de los puertos vienen acompañados de un *_c* para datos de captura y de *_e* para datos de esfuerzo. Este estilo de nombre se debe mantener.

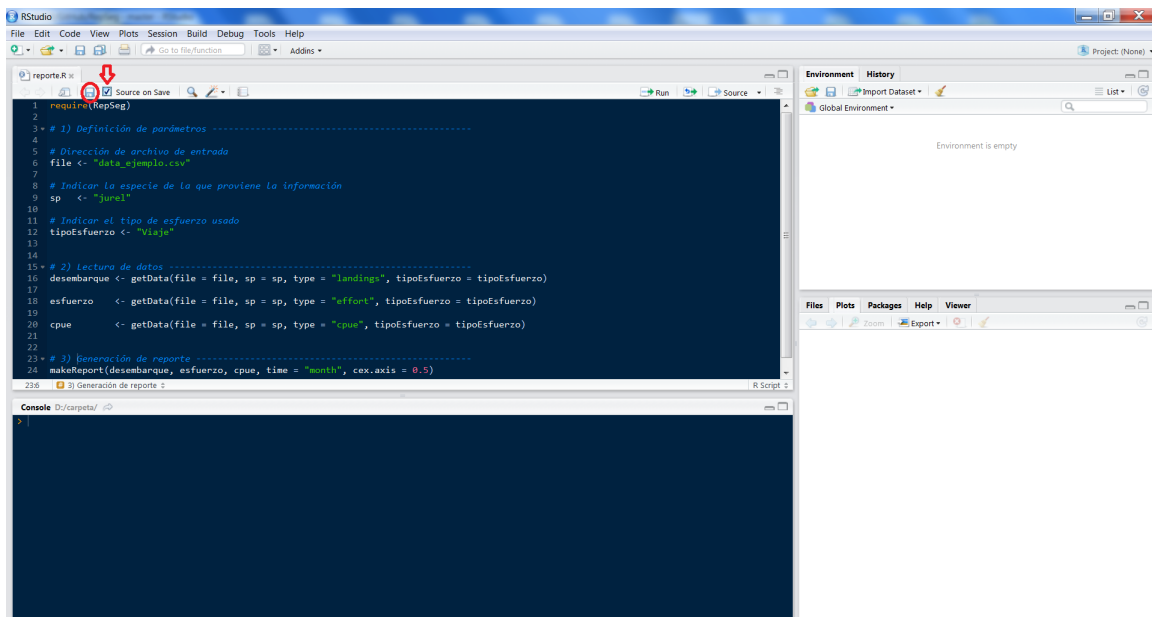
Así mismo, esta base de datos (p.e: *data_ejemplo.csv*) junto con el script para generar los reportes (p.e: *reporte.R*), deben estar guardados en una carpeta. Luego abrimos el archivo *reporte.R* (abrir con RStudio)



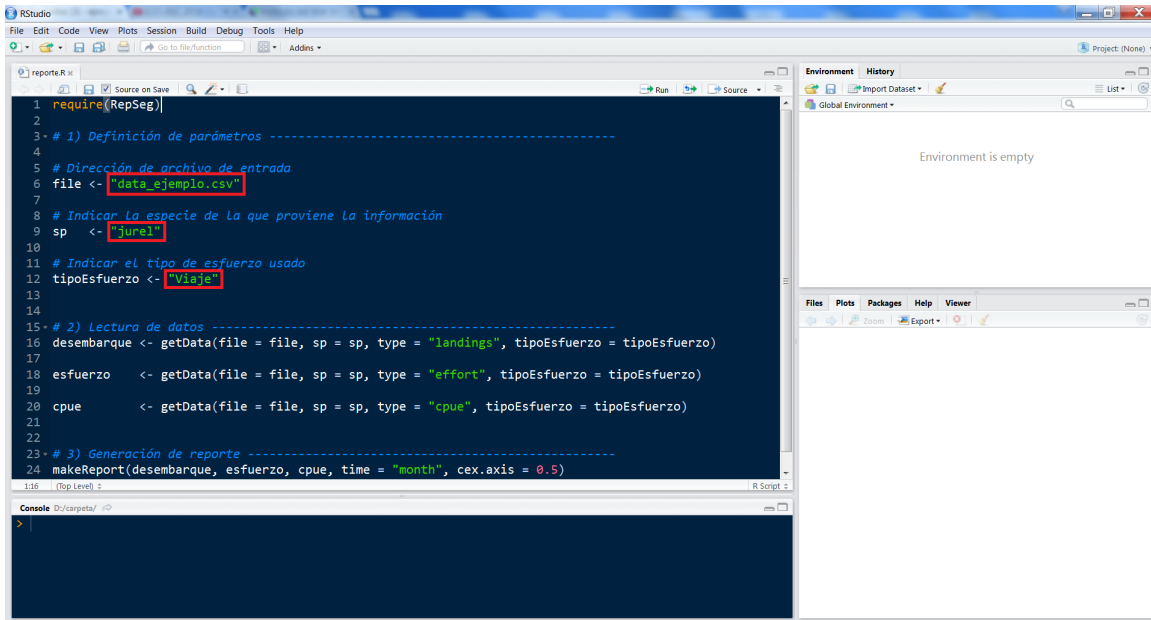
Una vez abierto el script, direccionar de la siguiente manera, haciendo click en **To Source File Location**:



Una vez realizados estos pasos, ya podemos obtener el reporte simplemente corriendo el script **reporte.R**. Para ello hacer check en **Source on Save** y darle click a la imagen del disquete como se muestra en la siguiente figura (círculo en rojo). Con estos pasos el script correrá y se visualizará el reporte.

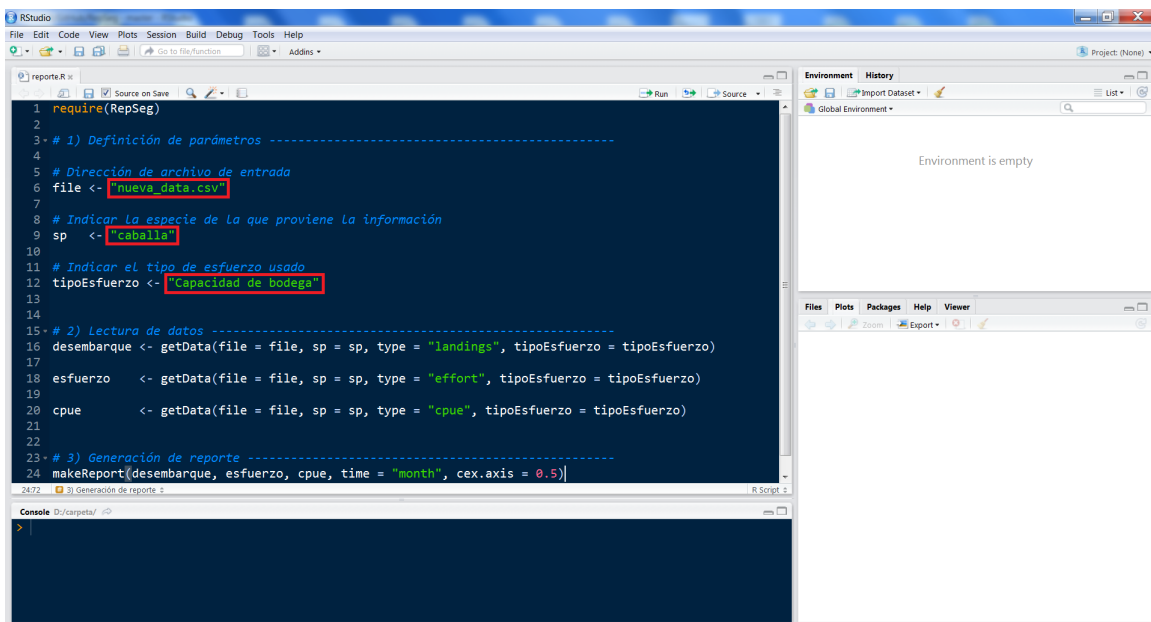


El script contiene 3 secciones: 1) *Definición de parámetros*, 2) *Lectura de datos* y 3) *Generación de reporte*. La sección de *Definición de parámetros* contiene: el nombre del archivo (*data_ejemplo.csv*), la especie sobre la cual se está haciendo el reporte (*jurel*) y el tipo de esfuerzo que contiene la base de datos (*Viaje*).



```
1 require(RepSeg)
2
3 # 1) Definición de parámetros -----
4
5 # Dirección de archivo de entrada
6 file <- "data_ejemplo.csv"
7
8 # Indicar la especie de la que proviene la información
9 sp <- "jurel"
10
11 # Indicar el tipo de esfuerzo usado
12 tipoEsfuerzo <- "Viaje"
13
14
15 # 2) Lectura de datos -----
16 desembarque <- getData(file = file, sp = sp, type = "landings", tipoEsfuerzo = tipoEsfuerzo)
17
18 esfuerzo <- getData(file = file, sp = sp, type = "effort", tipoEsfuerzo = tipoEsfuerzo)
19
20 cpue <- getData(file = file, sp = sp, type = "cpue", tipoEsfuerzo = tipoEsfuerzo)
21
22
23 # 3) Generación de reporte -----
24 makeReport(desembarque, esfuerzo, cpue, time = "month", cex.axis = 0.5)
```

Definición de parámetros es la única sección del script que el usuario cambiará, dependiendo del nombre del archivo, de la especie y el tipo de esfuerzo con que se trabaja.



```
1 require(RepSeg)
2
3 # 1) Definición de parámetros -----
4
5 # Dirección de archivo de entrada
6 file <- "nueva_data.csv"
7
8 # Indicar la especie de la que proviene la información
9 sp <- "caballa"
10
11 # Indicar el tipo de esfuerzo usado
12 tipoEsfuerzo <- "Capacidad de bodega"
13
14
15 # 2) Lectura de datos -----
16 desembarque <- getData(file = file, sp = sp, type = "landings", tipoEsfuerzo = tipoEsfuerzo)
17
18 esfuerzo <- getData(file = file, sp = sp, type = "effort", tipoEsfuerzo = tipoEsfuerzo)
19
20 cpue <- getData(file = file, sp = sp, type = "cpue", tipoEsfuerzo = tipoEsfuerzo)
21
22
23 # 3) Generación de reporte -----
24 makeReport(desembarque, esfuerzo, cpue, time = "month", cex.axis = 0.5)
```

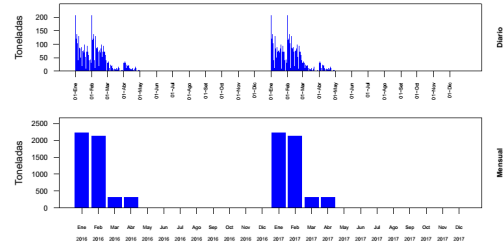


Nombre científico: *Scomber japonicus*
 Nombre común: Caballa
 Nombre en inglés: Chub mackerel
 Talla mínima de captura: 29 cm (LH)

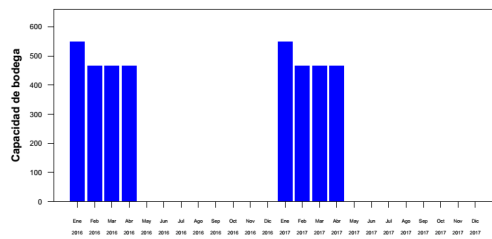
Desembarques (t)

Mes	2016	2017
Enero	2232	2232
Febrero	2129	2129
Marzo	322	322
Abril	322	322
Mayo	0	0
Junio	0	0
Julio	0	0
Agosto	0	0
Septiembre	0	0
Octubre	0	0
Noviembre	0	0
Diciembre	0	0
Total (t)	5006	5006

Desembarques



Esfuerzo pesquero



CPUE (t/Capacidad de bodega)

