

# Cómo obtener y visualizar datos satelitales de temperatura superficial del mar

## Ejemplo utilizando el paquete de R

## *rerddap*

Virginia A. García Alonso<sup>1,2,</sup> 

 @VA\_GarciaAlonso

 garciaalonso.v.a@gmail.com

Pablo S. Milla Carmona<sup>3,4,</sup> 

 @millacarmonia

 pabloomillac@gmail.com

<sup>1</sup> Universidad de Buenos Aires. Facultad de Ciencias Exactas y Naturales. Departamento de Biodiversidad y Biología Experimental. Buenos Aires, Argentina.

<sup>2</sup> CONICET-Universidad de Buenos Aires. Instituto de Biodiversidad y Biología Experimental Aplicada (IBBEA). Buenos Aires, Argentina.

<sup>3</sup> Universidad de Buenos Aires. Facultad de Ciencias Exactas y Naturales. Departamento de Ciencias Geológicas. Buenos Aires, Argentina.

<sup>4</sup> CONICET-Universidad de Buenos Aires. Instituto de Estudios Andinos “Don Pablo Groeber” (IDEAN). Buenos Aires, Argentina.

## Motivación

Las variables ambientales determinan muchos procesos biológicos en los ecosistemas marinos. La temperatura es especialmente importante en altas latitudes donde ocurren marcadas variaciones estacionales que influyen en los ciclos de vida de los organismos. Debido a los desafíos involucrados en obtener datos *in situ* y su baja resolución espaciotemporal, **utilizar datos satelitales emerge como una herramienta poderosa para impulsar los análisis marinos.**

## ¿Qué incluye este poster?

Un flujo de trabajo con los pasos para i) obtener **datos satelitales de temperatura superficial del mar (tsm)** desde el servidor ERDDAP usando el paquete *rerddap* (Chamberlain 2021) y ii) **visualizar** los resultados.

## ¿Dónde y cuándo?

El área de estudio abarca el extremo sur del Océano Atlántico Sudoccidental, donde existe una **marcada estacionalidad** y un **gradiente longitudinal** en la tsm entre temporadas. Visualizaremos datos de tsm de la primavera 2015 y otoño 2016.

## ¿Con cuál base de datos?

Con ‘jplMURSST41’, una base de datos de tsm de **ultra-alta resolución** generada en el *JPL Physical Oceanography DAAC (PODAAC)*. Esta incluye una **grilla global de 0,01 grados** con tsm interpoladas en grados Celcius (°C). Más información en este [sitio web](#).

Aquí puedes ver [otras bases de datos de ERDDAP](#).

## 1. Cargar los paquetes

Ademas del paquete *rerddap*, necesitaremos los paquetes *tidyverse* (Wickham 2019), *sf* (Pebesma 2022), *ggspatial*

(Dunnington 2021), *rnaturalearth* (South 2017a), *rnaturalearthdata* (South 2017b), *rnaturalearthhires* (South 2021) and *marmap* (Pante, Simon-Bouhet, and Irisson 2020) para manipular los datos y generar lindas visualizaciones :)

## 2. Descargar datos de tsm

Obtendremos datos de tsm con la función `info()` y eligiremos los datos deseados de acuerdo a la resolución espacio-temporal necesaria con la función `griddap()`.

```
#obtener base de datos de tsm
sstInfo <- info('jplMURSST41')

#elegir la resolucion espacio-
temporal
sst_spring <-

  griddap(sstInfo,
    latitude = c(-56, -52),
    longitude = c(-70, -55),
    time = c('2015-09-21',
             '2015-12-21'),
    fields = 'analysed_sst',
    fmt ="csv")
```

Descargar estos datos toma tiempo... para evitarlo, puedes guardarlos en un archivo `.csv` para tenerlos disponibles en el futuro, llamándolos con el comando

```
write.csv(sst_spring,"sst_spring.csv",row.names=F)
```

## 3. Leer y organizar los datos

Tenemos múltiples tsm para cada par latitud/longitud (una por día), por lo que estimaremos una **tsm promedio**.

```
#cargar los datos
sst_spring <-

  read.csv(file="sst_spring.csv",header=T)

#transformar y organizar los datos
mean_sst_spring <-

  #transformar en un data frame
  as.data.frame(sst_spring) %>%

  #seleccionar las variables
  dplyr::select(longitude, latitude,
                analysed_sst) %>%

  #estimar tsm promedio
  group_by(longitude,latitude) %>%

  summarise(mean_sst=mean(analysed_sst))
  %>%
  ungroup() %>%

  #generar nueva variable segun la
  temporada
  mutate(season="Primavera")
```

Repetimos el mismo procedimiento para **otoño**, creando un nuevo objeto llamado `mean_sst_autumn`, y finalizamos uniendo ambos conjuntos de datos en un único objeto.

```
#unir los datos de ambas estaciones
mean_sst <-
  bind_rows(mean_sst_spring,
            mean_sst_autumn)
```

## ¡Finalmente el gráfico!

Graficaremos los datos de tsm con *ggplot2*, usando la función `geom_raster()`.

```
ggplot()+

  #incluir la tsm como raster e
  interpolate
  geom_raster(data = mean_sst,
    aes(x = longitude,
        y = latitude,
        fill = mean_sst),
    interpolate = T)+

  #establecer el color de relleno
  scale_fill_viridis_c(alpha = 0.85)+

  #separar graficos por estacion
  facet_grid(season~.)
```

Algunos ajustes adicionales permiten incluir el territorio continental y contornos batimétricos relevantes al mapa, para obtener esto!

### Variabilidad estacional de la tsm

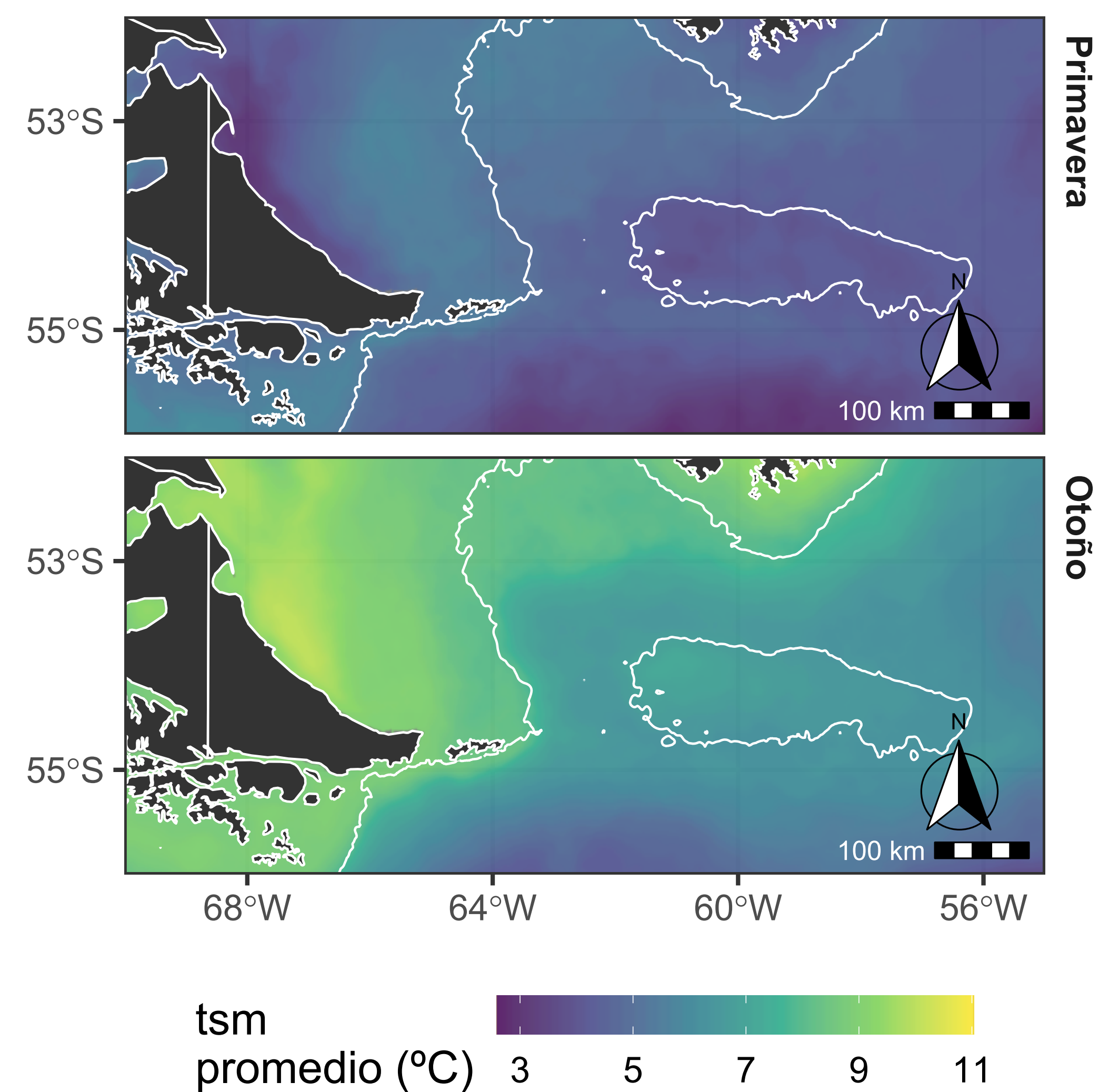


Figure 1: Temperatura superficial del mar (tsm) promedio en la primavera de 2015 y el otoño de 2016 en el área de estudio

Esperamos que esta explicación les haya resultado útil!!! Los materiales usados para este poster se encuentran a disposición en este [repositorio de GitHub](#).

## Referencias

Chamberlain, Scott. 2021. *Rerddap: General Purpose Client for ERDDAP Servers*. <https://CRAN.R-project.org/package=rerddap>.  
Dunnington, Dewey. 2021. *Ggspatial: Spatial Data Framework for Ggplot2*. <https://CRAN.R-project.org/package=ggspatial>.  
Pante, Eric, Benoît Simon-Bouhet, and Jean-Olivier Irisson. 2020. *Marmap: Import, Plot and Analyze Bathymetric and Topographic Data*. <https://github.com/ericpante/marmap>.  
Pebesma, Edzer. 2022. *Sf: Simple Features for r*. <https://CRAN.R-project.org/package=sf>.  
South, Andy. 2017a. *Rnaturalearth: World Map Data from Natural Earth*. <https://github.com/ropenscilabs/rnaturalearth>.  
———. 2017b. *Rnaturalearthdata: World Vector Map Data from Natural Earth Used in Rnaturalearth*. <https://github.com/ropenscilabs/rnaturalearthdata>.  
———. 2021. *Rnaturalearthhires: High Resolution World Vector Map Data from Natural Earth Used in Rnaturalearth*.  
Wickham, Hadley. 2019. *Tidyverse: Easily Install and Load the Tidyverse*. <https://CRAN.R-project.org/package=tidyverse>.