



PROYECTO:

NUEVOS INDICADORES DE DIVERSIDAD EN EL HÁBITAT PELÁGICO: RESPUESTA A LOS NUEVOS REQUERIMIENTOS PARA DEFINIR EL BUEN ESTADO AMBIENTAL PROPUESTO EN LA DIRECTIVA MARCO DE ESTRATEGIAS MARINAS (DIVERPEL)

GUÍA DE LA POTENCIAL APLICACIÓN DE LAS HERRAMIENTAS DISEÑADAS A LA EVALUACIÓN DEL ESTADO AMBIENTAL DE ZONAS PELÁGICAS EN EL MARCO DE LA DIRECTIVA MARCO DE ESTRATEGIAS MARINAS (DMEM)

[Proyecto aprobado por la Fundación Biodiversidad del Ministerio de la Transición Ecológica a través del programa pleamar 2018 y cofinanciado por el FEMP]

ÍNDICE

1. INTRODUCCIÓN.....	3
El Hábitat Pelágico.....	3
Fases planctónicas de los organismos marinos.....	4
Biodiversidad mar Mediterráneo.....	5
2. CASO DE ESTUDIO: EL MAR BALEAR.....	6
3. SERIES DE DATOS A LOS QUE APLICAR LAS HERRAMIENTAS.....	9
Tipo de datos	
Estructura de los datos (espacial y temporal)	
4. APLICACIÓN PASO A PASO DE LAS HERRAMIENTAS.....	11
Preparación bases de datos.....	11
Calculo diversidad Alfa.....	13
Análisis estadístico de las relaciones entre las variables ambientales y la biodiversidad.....	15
Determinación biorregiones.....	18
5. REFERENCIAS.....	22
ANEXO: SCRIPTS R Y VISUALIZADOR HERRAMIENTAS.....	24

1. INTRODUCCIÓN

HABITAT PELÁGICO Y DIRECTIVA MARCO

La reciente Decisión de la Unión Europea (2017/848) de incluir específicamente el Hábitat Pelágico (HP) así como ciertos criterios que deben considerarse para estimar su calidad siguiendo la Directiva Marco de Estrategias Marinas (DMEM, en particular en lo referente al Descriptor 1: Biodiversidad), ha dejado en evidencia la **existencia de un gran vacío metodológico y de conocimiento básico en lo que al sistema pelágico se refiere**. Por el momento, no existe consenso alguno en cuanto a la definición del Buen Estado Ambiental (GES, Good Environmental Status) para los HP. Sin embargo, resulta imprescindible determinar cuáles son las cualidades que un HP debe reunir para que se le considere en GES de tal forma que se puedan crear indicadores que permitan cubrir los objetivos de seguimiento y gestión propuestos en la DMEM y que resultan de obligado cumplimiento en la UE (Dickey-Collas et al. 2017).

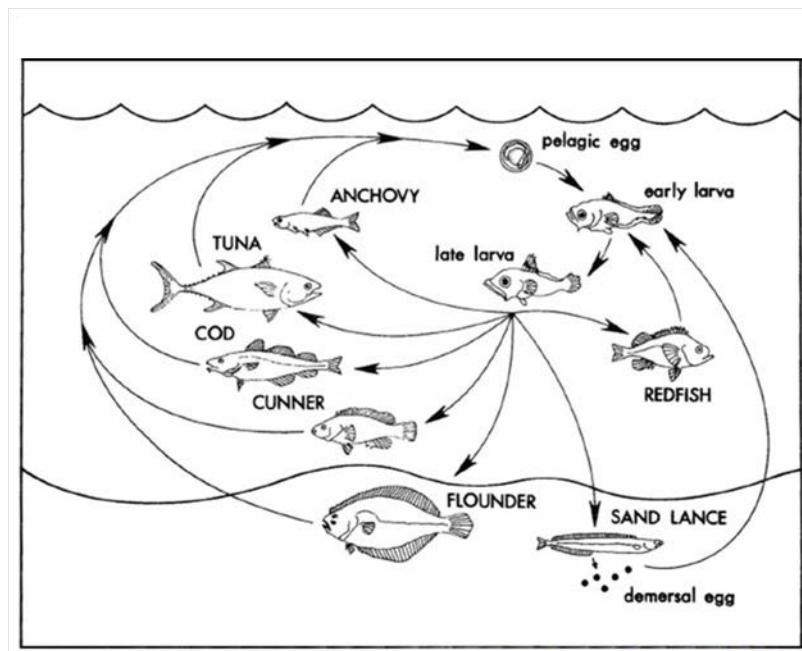
Si bien no hay consenso aún en cuanto a la definición del GES para los HP, si se han descrito las condiciones mínimas que debe cumplir el HP para cumplir unos criterios mínimos de GES. Una de esas condiciones que debe cumplir el HP es que todas las especies presentes en condiciones ambientales normales deben poder ser capaces de encontrar su HP necesario para completar su ciclo vital. Asimismo, deben mantenerse los roles globales regionales del sistema pelágico dentro de los ciclos biogeoquímicos. Otra de las condiciones que debe cumplir el HP para estar en GES es que no alteren la dinámica física del océano, ni por lo tanto a los movimientos de biota y masas de agua (Figura 1, Dickey-Collas et al. 2017). La principal diferencia entre el HP y los hábitats bentónicos o demersales radica en que se trata de un medio en el que la componente dinámica supera con creces a la estática tanto en los procesos físicos como en los biológicos por lo que definir el GES del HP resulta un **reto tanto a nivel de gestión, por el vacío existente** en la materia, como también **a nivel científico**, por el tipo de aproximaciones que la **dinámica del medio pelágico** exige.



Figura 1: Propiedades clave del estado del Hábitat Pelágico que contribuyen al Buen Estado Ambiental (GES) (Dickey-Collas et al. 2017).

FASES PLANCTÓNICAS DE LOS ORGANISMOS MARINOS

Las condiciones ambientales y biológicas del medio en el que habitan los organismos, tales como la temperatura, la salinidad o la clorofila, ejercen un papel fundamental en cómo se estructura la biodiversidad marina a través de gradientes biogeográficos (Coll et al 2010). Todos los seres vivos marinos, independientemente del hábitat que ocupen como adultos (costero o de aguas abiertas, pelágico, demersal o bentónico), están presentes en la columna de agua en alguno de los estadios de su ciclo vital. Eso se debe a que la mayoría de organismos marinos atraviesan una fase planctónica de huevo y/o larva que puede durar desde semanas a meses (Fig.2). En general, las tasas de supervivencia de las fases planctónicas de diferentes especies explotadas (peces o invertebrados) son muy bajas y muy variables dependiendo de las condiciones ambientales. Por ese motivo, las fases planctónicas son importantes tanto para determinar la estructura espacial de muchos recursos marinos en la fase dispersiva de su ciclo vital como por ser el periodo crítico en el que se determina el éxito de su proceso de reclutamiento y, en consecuencia, de la producción pesquera. Esto hace que el medio pelágico de una visión más completa de las especies que habitan en un ecosistema (i.e. biodiversidad) que si solo analizamos información obtenida en la fase adulta en el hábitats cercanos al fondo.



Miller and Kendall, 2009.

Figura 2: Organismos con hábitats diferentes en la fase adulta coinciden en la columna de agua en sus fases tempranas al ser éstas planctónicas (Miller and Kendall, 2009).

INDICADORES POTENCIALES BIODIVERSIDAD MARINA

El conocimiento del ambiente pelágico, de la características hidrográficas y dinámica de las masas de agua juega un papel fundamental para comprender los procesos que regulan tanto la dinámica de poblaciones explotadas por la pesca (Alvarez-Berastegui et al. 2014) como la diversidad marina (Hidalgo et al. 2015). Ese conocimiento resulta por tanto crucial para la evaluación y la planificación de medidas de gestión destinadas a asegurar la sostenibilidad de los recursos vivos pelágicos y demersales, así como planificar medidas de conservación a través de medidas de gestión espacial. Esto hace que la disponibilidad de series temporales de fases planctónicas y, variables hidrográficas y biogeoquímicas (nutrientes y oxígeno disuelto) constituyan una herramienta muy valiosa para seguimiento de biodiversidad marina y cambios en las masas de agua. Estos cambios se pueden utilizar para determinar indicadores y valores umbrales que definen el GES de un HP y su impacto en la biodiversidad. El hecho de utilizar datos de las fases planctónicas como indicadores de biodiversidad nos permite además ofrecer una **visión más completa de la biodiversidad marina** (excepto en el caso de los elasmobranquios, que no disponen de fase planctónica en su ciclo vital), ya que con un solo método de muestreo obtenemos organismos cuyos adultos proceden de muy diferentes hábitats (demersal, bentónico, mesopelágico o pelágico). Cualquier método de captura de ejemplares adultos resulta comparativamente más sesgado a la hora de establecer esa visión completa de GES que se persigue en la DMEM.

BIODIVERSIDAD MAR MEDITERRÁNEO

La biodiversidad del Mar Mediterráneo es muy alta, reflejando el amplio rango de condiciones climáticas e hidrológicas que permiten la supervivencia tanto de organismos de climas templados como de origen subtropical provenientes del océano Atlántico a los que hay que añadir el alto porcentaje de organismos endémicos mediterráneos (Coll et al 2010, Myers et al 2000). Dentro del Mediterráneo, el Mar Balear, se ha definido recientemente como "hotspot" o zona de alta incidencia de biodiversidad de organismos necto-bentónicos (Granger et al. 2015; Tugores et al 2018). Si bien no hay datos específicos respecto a la diversidad del hábitat pelágico, el área ha sido ya definida como de interés por ser su biodiversidad marina vulnerable a la presión de pesca de altura debido a que sus aguas albergan la principal zona de reproducción en el Mediterráneo de especies de túnidos y otros grandes migradores como es el caso del atún rojo (*Thunnus thynnus*) (UNEP-MAP RAC/SPA 2010). El hecho de que se haya definido la zona del Mar Balear como de alta biodiversidad en estudios independientes de especies pelágicas y el bentónico, sumado al hecho de que los sistemas insulares y en especial los de elevada actividad de mesoescala (corrientes, frentes y giros) favorecen la retención de los primeros estadios de vida de los organismos marinos (Bakun 2006, 2010) hacen del Archipiélago Balear un '*Laboratorio Ecológico*' único para investigar la biodiversidad en HP y las características que debe albergar para poder definir el GES.

2. CASO DE ESTUDIO: EL MAR BALEAR

En el caso del Mar Balear, el Instituto Español de Oceanografía lleva muestreando, de manera sistemática, los primeros estadios de las especies de vertebrados e invertebrados que se reproducen en primavera-verano desde el año 2001 hasta la actualidad (habiendo acabado recientemente la campaña de 2019), prestando especial atención a las especies de túnidos que se reproducen en la zona. Se trata de una serie de datos única en el mundo precisamente por esa atención especial que se dedica a la reproducción de los túnidos, que en el caso del atún rojo atlántico las zonas de reproducción son bastante restringidas y localizadas a nivel global (Mediterráneo y Golfo de México principalmente). Se trata de una malla de estaciones separadas 5 millas náuticas y con localización fija que se revisitan año tras año para visualizar, de manera sinóptica las comunidades de plancton asociadas a especies que se reproducen en primavera-verano y que rodean a las Islas Baleares (malla TUNIBAL, Fig. 3). Durante estos 19 años se ha conseguido muestrear de manera ininterrumpida gracias a diferentes fuentes de financiación y a bordo de diversos buques oceanográficos manteniendo la metodología para asegurar el estudio continuo de la serie temporal. En la actualidad la campaña ha pasado formar parte de la investigación estructural dentro del organigrama del IEO por lo que su continuidad en el tiempo queda asegurada. Además de los datos de las fases planctónicas de organismos marinos, la serie está compuesta por datos biogeoquímicos (nutrientes, oxígeno disuelto y clorofila) y datos relativos a la dinámica de las masas de agua (corrientes y estructuras de mesoscala como frentes o giros), por lo que resulta una serie de datos idónea para caracterizar el GES del HP en el Mar Balear. En la figura 3 se puede observar como ejemplo el área y las estaciones que se cubrieron en la campaña del año 2012.

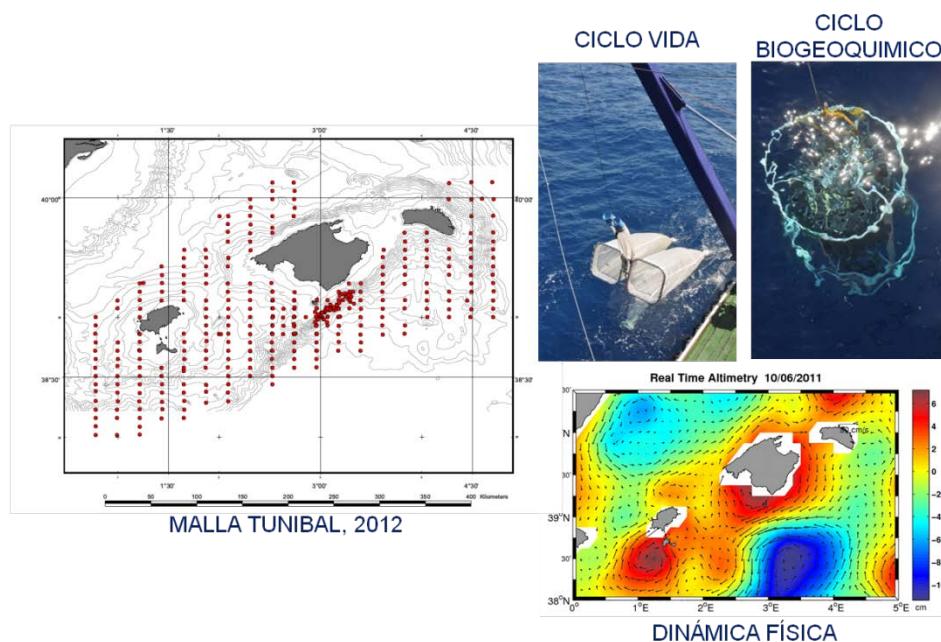


Figura 3: Mapa de las estaciones que se muestraron en el Mar Balear en el año 2012. En cada estación se obtuvieron datos relativos a fases planctónicas de organismos marinos, de los ciclos biogeoquímicos y de la dinámica física del sistema.

Como valor añadido, la serie de datos cubre casi todos los años gran parte de los espacios de especial protección reconocidos en Baleares como red Natura 2000 (Fig. 4) lo que permite evaluar también el GES en el HP de esas zonas. En ausencia de la dinámica física propia del océano cabría esperar valores diferentes de diversidad en estas áreas de especial protección en comparación con los valores de aquellas zonas que carecen de protección. El hecho de que si existe esa dinámica física implica que debe estudiarse detenidamente si en estas áreas la dinámica favorece, desfavorece o no tiene influencia ninguna sobre la diversidad de organismos marinos.

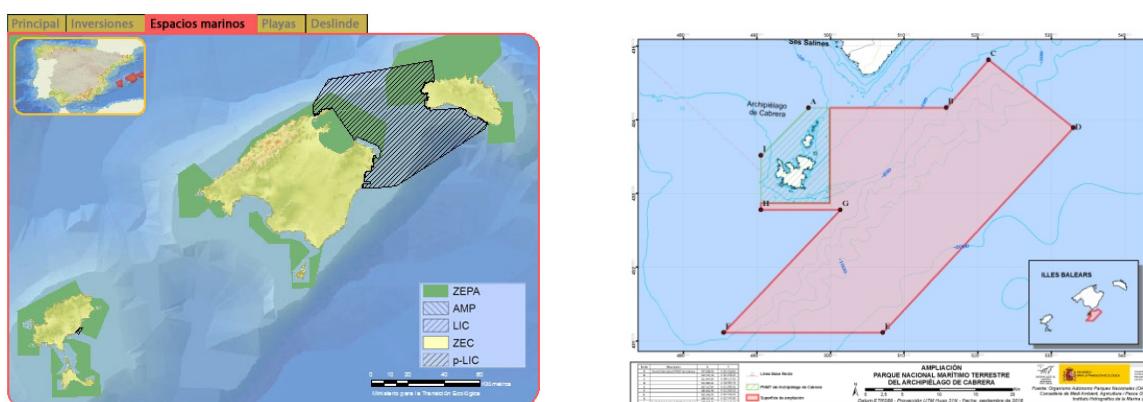


Figura 4: Izquierda: Mapa de las áreas de la Red natura 2000 en Baleares. Derecha: Detalle de la reciente ampliación del parque nacional Marítimo-Terrestre del Archipiélago de Cabrera (www.miteco.gob.es).

Todo esto hace que, a día de hoy, nos encontramos frente a un doble reto (de gestión y científico) para el que tenemos que desarrollar herramientas y nuevo conocimiento necesarios para afrontarlo.

a) Retos:

En términos de **gestión** la DMEM es de obligado cumplimiento por los estados miembros y aún no existe una metodología clara de cómo determinar el buen estado ambiental del hábitat pelágico. No sólo se debe establecer el GES del HB sino que se deben establecer rangos y umbrales tanto de la diversidad como de las posibles presiones a las que se vea expuesta (la Decisión UE 2017/848 propone: "*El establecimiento de valores umbral debe tener en cuenta la naturaleza dinámica de los ecosistemas marinos y sus elementos, que pueden cambiar en el espacio y el tiempo debido a las variaciones hidrológicas y climáticas, a las relaciones predador-presa y a otros factores medioambientales. Los valores umbral deben reflejar asimismo el hecho de que los ecosistemas marinos pueden recuperarse, si estuvieran deteriorados, a un estado que refleje las condiciones fisiográficas, geográficas, climáticas y biológicas predominantes, en lugar de volver a un estado anterior concreto*".

En términos **científicos** la dinámica inherente al reino pelágico añade complejidad a cualquier aproximación a modelar el sistema que se pretenda acometer. Hay que desentrañar cual es el papel de las estructuras oceanográficas y su alto dinamismo sobre la variaciones a corta escala (local) y

media (regional) en la biodiversidad pelágica. El largo recorrido temporal de la serie va a permitir también identificar la variabilidad interanual del sistema. También resulta interesante el hecho de obtener una visión completa de la biodiversidad, tanto la taxonómica (p.e. índices de diversidad clásicos como riqueza específica) como la funcional (p.e. aquellos índices que tiene en cuenta características vitales de los organismos directamente relacionadas con la función que desempeñan dentro del ecosistema) (Granger et al. 2015).

b) Herramientas:

La disponibilidad de una serie de datos de plancton con una componentes espacial como es el caso de TUNIBAL es algo único a nivel nacional, la cual nos va a permitir no solo determinar la biodiversidad del Mar Balear en un momento determinado sinocompararla a lo largo del tiempo y en relación a presiones tanto de origen antropogénico (p.e. variaciones en la concentración de oxígeno disuelto o nutrientes) como de origen natural (p.e. procesos oceanográficos regionales). El largo recorrido temporal de la serie nos va a permitir identificar rangos y valores umbral para la mayoría de las presiones así como su patrón espacial, lo que añade además la posibilidad de diferenciar entre las zonas sin protección y las especialmente protegidas.

Los resultados y las herramientas de análisis del presente proyecto serán pues exportables a otras zonas del Mediterráneo, e incluso del Atlántico, para la evaluación del GES de sus HP en las que existen series de datos similares a la que el IEO posee en el Mar Balear.

3. SERIES DE DATOS A LOS QUE APLICAR LAS HERRAMIENTAS

TIPO DE DATOS

Las herramientas para calcular el estado ambiental del hábitat pelágico pueden utilizarse con cualquier base de datos de organismos pelágicos. Puede tratarse de datos fitoplancton, zooplancton (micro, meso y ,macro) y dentro del zooplancton el ictioplancton (larvas de peces).

La serie de datos debe estar constituida por datos comparables entre sí es decir, el método de extracción de organismos y el esfuerzo de pesca empleado debe haber sido el mismo a lo largo de toda la serie y en el caso de que no haya sido así deben haberse realizado ejercicios de calibración entre métodos diferentes.

La identificación de taxones debe también ser equiparable entre los datos de la serie. Cuando mas de un taxónomo ha intervenido en las identificaciones debe asegurarse que todos los datos se refieren al mismo nivel taxonómico. Cuanto mas individuos se identifiquen a nivel especie mas sensible será el resultado del uso de las herramientas a las diferencias de diversidad.

Si la serie de datos planctónicos viene acompañada por una serie de datos físico-químicos podrá determinarse además el efecto que ejercen las presiones tanto ambientales como antropogénicas sobre la biodiversidad en el hábitat pelágico.

ESTRUCTURA DE LOS DATOS

A) ESPACIAL

La estructura espacial de los datos debe ser la suficiente como para permitir detectar diferencias entre estructuras naturales. En el caso del HP esas estructuras tienen en general que ver con la dinámica marina y las peculiaridades del lecho marino.

Si el interés de la determinación del estado ambiental está relacionado con presiones antropogénicas, la serie de datos deberá cubrir todo el área de posible influencia de dichas presiones.

Es obvio decir que la serie de datos planctónica y las de presiones deben poseer la misma estructura espacial para utilizar las herramientas que proponemos

B) TEMPORAL

La estructura temporal de los datos debe ser la suficiente como para permitir detectar diferencias entre tiempos determinados. Si se trata de especies de puesta de primavera, todas las muestras de la serie deberán tomarse en primaveras consecutivas. En el caso de querer determinar diversidad a lo largo del año, la toma de muestras deberá espaciarse lo suficiente como para detectar cambios asociables a la temporalidad (en el caso de zonas templadas la estacionalidad puede ser una fuente de diferencias de diversidad).

Si el interés de la determinación del estado ambiental está relacionado con presiones antropogénicas, la serie de datos deberá cubrir el tiempo en el que se ha ejercido la presión. En el caso de presiones puntuales en el tiempo, los datos de antes, durante y tras la presión pueden ser muy reveladores.

Es obvio decir que la serie de datos planctónica y las de presiones deben poseer la misma estructura temporal para utilizar las herramientas que proponemos.

4. APLICACIÓN PASO A PASO DE LAS HERRAMIENTAS

Como anexo a esta guía se presenta la herramienta creada en el marco del proyecto DIVERPEL: los scripts en el lenguaje de programación R (software libre) para la realización de los cálculos necesarios para determinar los índices de diversidad y su relación con parámetros ambientales que permitan determinar el estado ambiental del HP a través de series sistemáticas de datos de organismos planctónicos.

El paso a paso de cómo ejecutar dichos scripts:

PREPARACIÓN BASES DE DATOS (Abu_data.R)

En nuestro Caso de Estudio, la investigación sistemática de los primeros estadios del ciclo de vida de las especies de peces en las Islas Baleares se remonta al año 2001. Se trata de una serie de campañas anuales enfocadas al estudio de la reproducción de los túnidos alrededor del archipiélago balear. La época de puesta de los túnidos coincide con la de muchas otras especies de peces que habitan estas aguas o las utilizan para su reproducción por lo que coexisten en el pláncton. En las campañas se recolectaban tanto larvas de peces (y otros organismos presentes en el plancton) como muestras de agua y datos de temperatura, salinidad y oxígenos disuelto que permiten caracterizar el ambiente en el que las larvas de peces se están desarrollando. Las muestras se tomaban año a año en una red sistemática de estaciones separadas generalmente 10 millas náuticas entre sí. Algunos años se realizaron muestreos en otras estaciones, siguiendo diferentes objetivos como por ejemplo el estudio de la deriva por viento y corriente de una mancha detectada de larvas de atún. En la figura 2.1 quedan reflejadas las estaciones que se han muestreado entre los años 2001 y 2019.

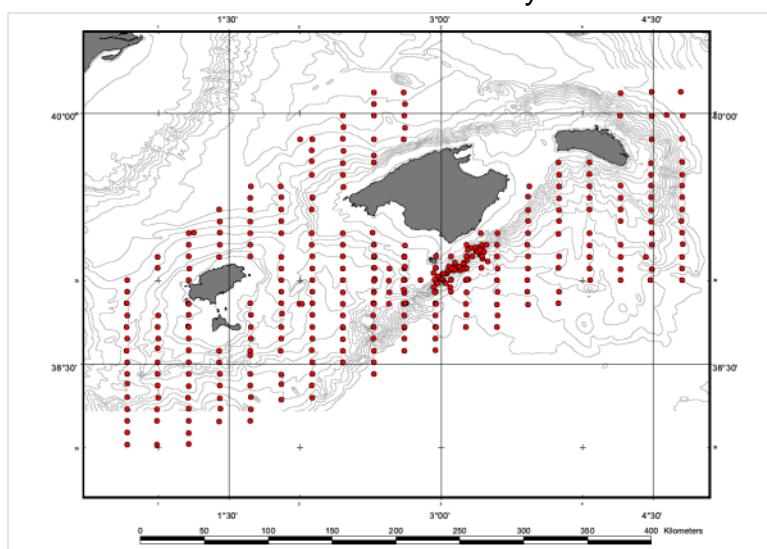


Figura 2.1_ Red sistemática de estaciones de la malla TUNIBAL. En vertical están representadas las estaciones cada 5 millas náuticas y en la horizontal cada 10. Se observa como la zona del Parque nacional marítimo-terrestre del archipiélago de Cabrera suele recibir mayor intensidad de muestreo.

A lo largo de los años, la serie de datos se ha ido completando con la financiación de diferentes proyectos.

Entre los años:

- 2001-2003 las campañas se denominaban TUNIBAL (Túnidos de Baleares)
- 2004-2005 Campañas TUNNIBAL II
- 2006-2010 Campañas MINITUNIBAL
- 2011- 2017 Campañas BLUEFIN
- 2018- indefinido Campañas TUNIBAL PNBD (Se convierte en campaña estructural del Instituto Español de Oceanografía, lo que supone que la serie está asegurada como prioridad del organismo)

Año a año, a medida que el conocimiento de la especie iba avanzando, se iban modificando las técnicas de captura para adaptarlas a las especies objetivo (túnidos). Así, una vez descubierto que este tipo de larvas habitaba casi exclusivamente en los primeros metros de la columna de agua, se pasó a aumentar el esfuerzo de muestreo en esa primera capa de agua. Por lo tanto, a la hora de enfrentarnos a la base de datos hubo que tomar ciertas decisiones para que los resultados fueran lo más aproximados a la realidad posibles.

Así, para el presente trabajo se decidió utilizar solo los datos de aquellos años en que la metodología de trabajo fue estrictamente igual y esos datos, por lo tanto comparables entre sí sin necesidad de calibraciones o transformaciones que puedan sesgar los resultados.

Los datos que se escogieron son los que comprenden el periodo 2012-2016 (5 años). Los datos posteriores a 2016 no tenían, en el momento de comenzar el proyecto, la calidad suficiente para ser incluidos.

Sobre la base de datos de esos 5 años se realizaron las siguientes actuaciones:

- Comprobar duplicados
- Eliminar columnas con 0's
- Eliminar datos raros
- Juntar columnas con misma especie pero distinto código
- Juntar abundancias de túnidos e ictioplancton
- Calcular densidad de larvas en 100 m³ para cada uno de los taxones identificados

Por lo tanto, partimos de una base de datos de 650 observaciones (las diferentes estaciones de muestreo en los diferentes años) de 135 taxones de larvas de peces (5 de ellos de especies de túnidos).



Con los datos ya depurados se creó una matriz de taxones con los datos de densidad .

A la hora de proponer índices que caractericen la diversidad en la zona, con los datos de los que disponemos, habrá que primar aquellos en los que la abundancia proporcional estudia la equidad (toman en cuenta el valor de importancia de cada especie) frente a los de dominancia (tienen en cuenta la representatividad de las especies con mayores valores de abundancia sin evaluar la contribución del resto de especies)

CALCULO DIVERSIDAD ALFA (Biodiversity.R)

Comenzamos por estudiar la diversidad de lo que consideramos la comunidad del hábitat pelágico del Mar Balear en su conjunto. Para obtener parámetros completos de la diversidad de especies en un hábitat debemos cuantificar el número de especies y su representatividad.

1. Riqueza específica (número de especies)
2. Estructura de la comunidad (distribución proporcional del valor de importancia de cada especie, en nuestro caso abundancia relativa de cada especie). Esta estructura puede analizarse teniendo en cuenta la equidad y la dominancia, antes mencionadas. En nuestro caso usamos los análisis del Índice de Simson (dominancia) y Shannon (equidad).

Las fórmulas utilizadas para el cálculo:

- Riqueza (numero de especies en cada estacion)

$$S = \text{nº especies}$$

- Dominancia (índice de Simpson Dsi, 1949)

$$D_{Si} = \sum_{i=1}^S p_i^2$$

p_i = abundancia proporcional de la i-ésima especie; representa la probabilidad de que un individuo de la especie i esté presente en la muestra, siendo entonces la sumatoria de p_i igual a 1

Se calcula:

$$p_i = \frac{n_i}{N}$$

n_i = número de individuos de la especie i

N= número total de individuos para todas las S especies en la comunidad

- Equidad (Índice de Shannon-Wiener (Shannon y Weaver, 1949), H')

$$H' = - \sum_{i=1}^S (p_i \times \log_2 p_i)$$

H'= índice de Shannon-Wiener que en un contexto ecológico, como índice de diversidad, mide el contenido de información por individuo en muestras obtenidas al azar provenientes de una comunidad 'extensa' de la que se conoce el número total de especies S.

El cálculo se realizó para todas las estaciones todos los años.

Todos los cálculos se realizaron para el set de datos completo y separando túnidos del resto: la separación no aporta mucho así que a partir de ahora trabajaremos con el set completo.

El índice de Simpson (recordemos, aquel en el que el peso de las especies mas abundantes acapara toda la variabilidad) no aporta demasiada información cuando lo comparamos entre años. Al estar representado 1-Simpson, cuanto más se acerca el valor de D a 1, mayor es la diversidad del hábitat. Se trata de las zonas más rojas en los mapas.

El índice de Shannon está basado en un concepto muy relevante en ecología: la uniformidad. Este parámetro hace referencia al grado en el que las especies están representadas a lo largo de la muestra. Los extremos abarcan una sola especie dominante y otras especies presentes en número muy bajos (valores de uniformidad cercanos al 0), a todas las especies representadas por números iguales (valores de uniformidad cercanos al 1). La uniformidad tiene un papel fundamental en el análisis ecológico de la diversidad. Por ejemplo, en comunidades más uniformes, el índice de Shannon se vuelve más sensible a la riqueza.

En los mapas del Mar Balear, las zonas de mayor riqueza específica coinciden con los mayores valores de Shannon, por lo que podemos decir que es un índice que va a representar bien la diversidad a la hora de estudiar sus cambios a través de gradientes (diversidad beta).

Con la última parte del script se ha querido mostrar gráficamente la diferencia entre incluir todo el set de datos de larvas de peces (complete) o estudiar de manera separada la diversidad de túnidos (especies a las que el muestreo iba originalmente dirigido) del resto de la comunidad de larvas de peces. El peso de la diversidad de los túnidos en la diversidad total en el caso del Mar balear es muy bajo, pese a ser *T. thynnus* la especie mas abundante con diferencia ya que hemos escogido un índice en el que se suavicen las especies dominantes para poder cambiar cambios en las menos abundantes.

Cambios que pueden suponer cambios drásticos en los ecosistemas a nivel más local.

ANÁLISIS ESTADÍSTICO DE LAS RELACIONES ENTRE LAS VARIABLES AMBIENTALES Y LA BIODIVERSIDAD (myCTD.R, DIV+CTD.R, HOM_DATA.R, DIV+CTD+DENS.R, SHANNONGam_200224.R)

myCTD.R

De la misma manera que con los datos de larvas de peces. La primera tarea dentro de esta acción fue la preparación de la base de datos de variables ambientales

Los datos iniciales disponibles provenientes de sonda CTD para los años seleccionados (2012-2016) se resumen en la siguiente tabla:

Year	Operations	Stations	Replicates
2012	183	168	12
2013	168	158	9
2014	112	111	1
2015	92	92	0
2016	95	95	0

Se eliminaron tanto los datos replicados como los que provenían de muestreos nocturnos, ya que estos sólo se realizaron en el año 2002.

Las variables incluidas en la base de datos, teniendo en cuenta los valores obtenidos con ctd y las características de la estación, la campaña y la pesca fueron:

- Campaña (survey)
- Año (year)
- Estación, Latitud, Longitud (estación, lat, lon)
- Hora del día de la pesca (TIME)
- Número de operación dentro de la campaña (operation)
- Profundidad de la estación (st_depth, ST_Z)
- Profundidad del perfil (min_depth)
- Profundidad de la capa de mezcla en la estación (mld)
- Duración de la pesca (tow_duration)
- Oxígeno en la capa de mezcla, 200m, 100m, 50m, 25m, 10m, 5m (omld ,ox200, ox100, ox50, ox25, ox10, ox5)
- Temperatura en la capa de mezcla, 200m, 100m, 50m, 25m, 10m, 5m (tmld ,temp200, temp100, temp50, temp25, temp10, temp5)
- Salinidad en la capa de mezcla, 200m, 100m, 50m, 25m, 10m, 5m (smld ,sal200, sal100, sal50, sal25, sal10, sal5)
- Modulo y velocidad de la corriente geostrófica (gvel_u, gvel_v)
- Clorofila en la capa de mezcla (media y acumulada), la media y acumulada entre 10 y 100m, el máximo, la profundidad del máximo, la mensual

y la semanal (mean_fluor_mld, cumul_fluor_mld, mean_fluor_10_100, cumul_fluor_10_100, chl_month, chl_week)

Los filtros y comprobaciones que se realizaron para obtener la base de datos ambiental se presentan en el script. Se realiza un gráfico en el que se muestra el numero de observaciones de cada variable que se disponía. Se observa como si no se dispone de datos de ciertas variables para determinadas campañas e incluso para la mayoría de las observaciones deben desecharse las variables con mayor falta de dato (sensores que no funcionan correctamente por ejemplo...).

A continuación se representa espacialmente la posición de las observaciones cada uno de los años. En nuestro caso, tras observar que en la campaña de 2012 se cubrieron muchas mas estaciones al norte de la isla de Mallorca, se retiraron esas estaciones ya que no permitirían la comparación entre años.

SHANNONgam_200224.R

A continuación, para analizar estadísticamente las relaciones entre las variables ambientales y la biodiversidad se usó las metodología propuestas en Hidalgo et al. (2014). Se trata de aplicar Modelos Aditivos Generalizados (GAMs) para analizar la relación funcional entre las variables ambientales y la biodiversidad pelágica. Como índice de diversidad se utilizó el Shannon, para que la presencia de las especies más abundantes no enmascarase los cambios de diversidad. A diferencia del Modelo Lineal Generalizado, que restringe la relación entre la variable respuesta y la explicativa a la forma lineal, el Modelo Aditivo Generalizado permite que las funciones puedan tomar cualquiera forma funcional, lo que proporciona más información acerca de la relación entre la variable explicativa y la variable objetivo.

De manera sencilla, lo que se busca aplicando el GAMs es conseguir una ecuación de tipo:

$$\text{Shannon(diversidad)} \sim \text{varambiental1} + \text{varambiental2} + \dots + \text{varambientaln}$$

En nuestro caso, teniendo en cuenta que las pescas de larvas de peces se realizaron siempre sobre la capa de mezcla, no se intentó relacionar la diversidad con ninguna variable a profundidades mayores a dicha capa de mezcla. Dada la estabilidad de la columna de agua en esta parte del Mediterráneo en verano (fuerte estratificación) se escogieron las variables que representaban la capa de mezcla, ya que dicha capa está íntimamente relacionada tanto con procesos físicos como químicos y biológicos.

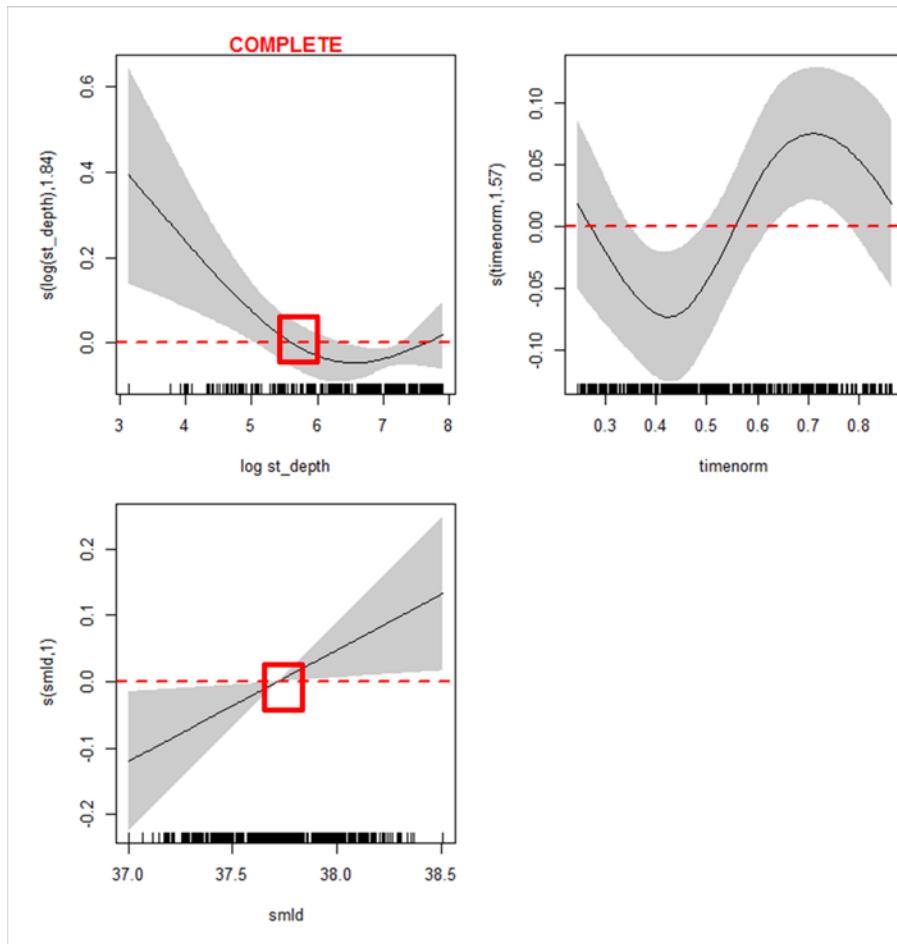
Se comenzó enfrentando el índice de diversidad de Shannon a variables de posición (latitud/longitud), profundidad de la estación (determina en esta zona la distancia a costa), y variables mas dinámicas como la hora del dia a la que se tomó la muestra, la salinidad, la temperatura etc...

Se realizaran diferentes modelos con diferentes combinaciones de variables ambientales. Se confrontaron dichos modelos hasta encontrar el que mejor explicaba la variabilidad en los cambios de diversidad. En nuestro caso, el modelo resultante:

$$\text{Shannon} \sim \text{ST_Z}^{**} + \text{TIME}^{**} + \text{SMLD}^*$$

Lo que significan que una combinación de la profundidad de la estación, la salinidad y la hora del día a la que se pescó pueden explicar una parte de la variabilidad en las diferencias de diversidad.

Con el script podemos representar este resultado, que en nuestro caso fue:



La respuesta de la hora del día (timenorm) está relacionada con los ritmos circadianos de los organismos y queda claro que hay dos veces al dí que se capturan mas larvas de peces (las dos veces que la curva cruza la línea recta que pasa por el 0 “efecto”). En cuanto a la profundidad de la estación (st_depth). El resultado nos muestra como va cayendo la diversidad a medida que va aumentando la profundidad del agua es decir, a medida que pasamos de un sistema costero a uno de aguas mas abiertas. Finalmente, el resultado tan claro (lineal) de la salinidad (smld), muestra como la diversidad aumenta a

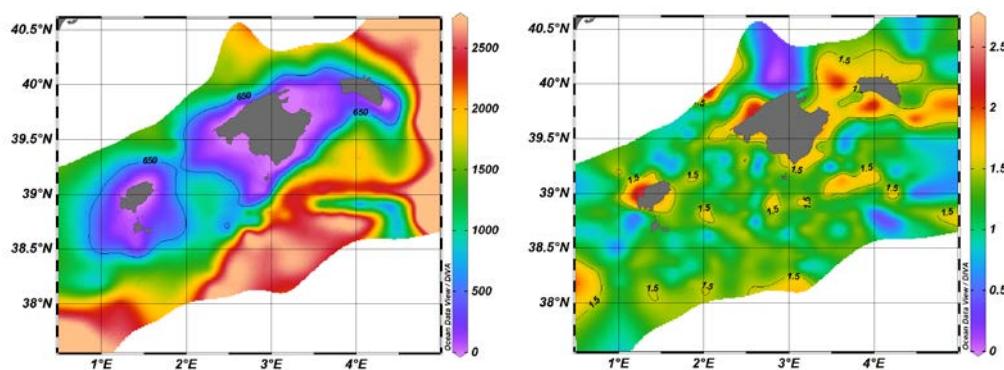
medida que lo hacen las aguas salinas. En el caso de esta zona del mediterráneo en esta época del año (verano), la salinidad puede usarse como determinante del tipo de agua. Las aguas mas salinas en la zona son las que llevan meses en el Mediterráneo circulando y han ido evaporándose a lo largo del camino. Las aguas menos salinas corresponde al agua nueva del Atlántico, que ha entrado recientemente en el Mediterráneo. La zona intermedia corresponde a la transición entre ambos tipos de aguas. Se trata de las conchas de frente, convergencia etc...

A partir de estos resultados se procedió a la identificación de biorregiones. En el caso de la profundidad de la estación, el punto de inflexión de la curva diversidad-efecto parcial de la profundidad se encuentra en los 650m (en un recuadro rojo en el gráfico). En cuanto a la salinidad, el valor a partir del cual se aprecia un cambio en el efecto parcial sobre la diversidad es 37.7.

MAPAS BIORREGIONES: VISOR (DIV_MAPS.R)

A partir de los resultados obtenidos aplicando el método de los GAMs, se procedió a identificar las diferentes biorregiones y los posibles indicadores de la diferencia de diversidad entre ellas. Para ello, en el caso de la profundidad, se estudió la composición de especies de las zonas con profundidades mayores de 650m y de las de profundidades menores de manera separada. Se compararon ambos grupos después mediante un análisis SIMPER (porcentaje de similitud) para identificar cuales eran las especies que caracterizaban cada uno de los grupos y cuales caracterizaban sus diferencias (estas últimas podrían considerarse como indicadores).

MAPA DE BIORREGIONES BASADO EN DIFS DEPTH GAM: ~ 650m



Isolinea profundidad en los 650 m que separa las biorregiones identificadas tras analizar el efecto parcial de la profundidad sobre el índice Shannon de diversidad
Isolinea índice de Shannon en el valor de 1.5 para enfatizar la relación con la profundidad que caracteriza las biorregiones identificadas

Se presenta el mapa de las Biorregiones según la profundidad. A su lado el de diversidad, para ayudar a la interpretación del modelo GAM arriba

presentado. En nuestro modelo, se identifica la profundidad de 650 m como el límite a partir del cual se notan diferencias en la biodiversidad relacionados con la profundidad. En el caso del Archipiélago Balear, teniendo en cuenta su orografía, las zonas menos profundas coinciden con las más cercanas a costa e incluyen al canal de Menorca. La observación conjunta de ambos mapas permite observar como las zonas en las que la diversidad es mayor son en general las más cercanas a costa. Probablemente se deba al mayor número de especies ya que en estas zonas se suman las especies más "locales" cuyo hábitat adulto es el costero con lo que realizan la puesta muy cerca con especies más "migrantes" o de hábitat más extendido que aprovechan las características hidroclimáticas de estas aguas para la puesta y asegurar la supervivencia larvaria. A continuación en el mismo anexo presentamos los resultados del análisis realizado por separado a las poblaciones que caracterizan las dos biorregiones (<650m y >650m). El análisis de SIMPER (porcentajes de similitud) realizado para cuantificar la contribución de las especies en las biorregiones observadas calcula el porcentaje de diferencia en la composición por especies de dos o más grupos pre-establecidos. Si dicho porcentaje supera el 70% podemos suponer que los grupos son diferentes. En el caso de las biorregiones basadas en la profundidad, la diferencia entre grupos encontrada fue del 73%. En el Anexo II se presenta, en forma de lista, la contribución de cada especie o taxón a la diferenciación de dichos grupos. En la Biorregión menos profunda, a excepción del Atún Rojo (*Thynnus thynnus*) que está presente en todos los ambientes, las especies que más contribuyen en la caracterización y diferenciación de la biorregión son de tipo costero (marcadas en azul). Es el caso del boquerón (*Engraulis encrasicolus*), la tuta (*Chromis chromis*), la doncella (*Coris julis*) y la familia de los Gobios entre otros. Las especies que caracterizan la biorregión más profunda son las que el adulto es mesopelágico (habita desde los 200 a los 4000 m de profundidad y suele realizar migraciones verticales para su alimentación). Se trata de especies de mictófidos (*Ceratoscopelus maderensis*, *Hygophum spp.*, *Lampanyctus spp* etc..) y gonostomátidos (*Cyclothona pygmaea* y *Cyclothona braueri*). En el caso de los túnidos observamos como el atún rojo, si bien es un gran migrador cuyos adultos habitan aguas más abiertas, caracteriza la comunidad de la biorregión menos profunda. La melva (*Auxis rochei*), de hábitat bastante más costero, aparece como especie más característica de aguas poco profundas mientras que el atún blanco (*Thunnus alalunga*) también gran migrador aparece caracterizando la comunidad de la biorregión más profunda.

Por lo tanto, para estudios posteriores de la biodiversidad proponemos el estudio del cambio de la diversidad con la profundidad. Cualquier diferencia significativa respecto a los 650 m observados en esta serie, deberá ser objeto de análisis para determinar variables que hayan podido influir en esa variación (recordemos que la profundidad es una variable estática, probablemente nunca

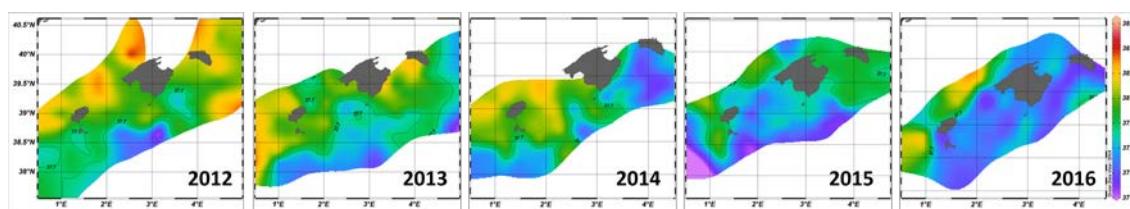
observaremos grandes cambios, no observaremos cambios en la profundidad a los que asociar un cambio en la diversidad).

En cuanto a la respuesta de la diversidad frente a los cambios en la salinidad de la capa de mezcla, el valor umbral que podemos extraer del análisis GAM es de 37.7.

MAPAS DE BIORREGIONES BASADO EN DIFERENCIAS SALMLD GAM: ~ 37.7

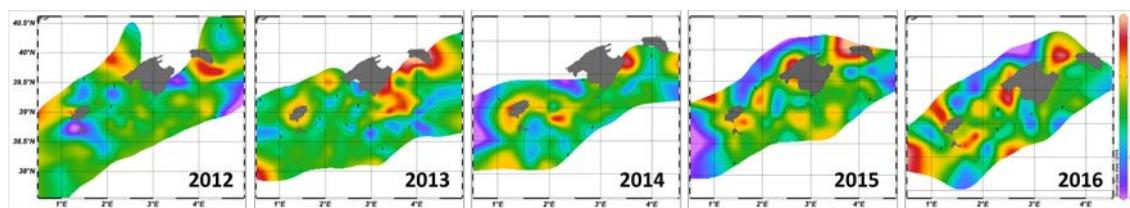
a) BIORREGIONES SEGÚN SALINIDAD EN LA CAPA DE MEZCLA

Isolinea en los 37.7 que separa las biorregiones identificadas tras analizar el efecto parcial de la salinidad en la capa de mezcla sobre el índice Shannon de diversidad



b) INDICE DE SHANNON OBSERVADO CADA AÑO

Isolinea en el valor de 1 para enfatizar la relación con la SALINIDAD EN LA CAPA DE MEZCLA que caracteriza las biorregiones identificadas



Es importante destacar, en lo que a la salinidad de sefiere, que se trata de un parámetro dinámico en comparación con la profundidad. Eso significa que cada año, cada una de las estaciones de muestreo puede presentar valores de salinidad diferentes a años anteriores o posteriores. Por este motivo presentamos en el Anexo IIh los mapas de las biorregiones para cada uno de los años. Hay que destacar también que en el noroeste del Mediterráneo, la salinidad se utiliza para caracterizar las masas de agua y es precisamente alrededor del valor de 37.7 donde se establece la diferencia entre la masa de agua mas fresca o reciente, que proviene del atlántico (menos salina) y la masa de agua que ya lleva un tiempo en el Mediterráneo, y en su recorrido ha ido evaporándose y, por lo tanto aumentando su salinidad. En función de que el

invierno en el Golfo de León haya sido mas o menos frío, se forma una masa de agua mas densa que va avanzando hacia y en verano el sur ocupa los canales de Menorca y Mallorca no permitiendo el avance del agua nueva hacia el norte. Por esta razón presentamos el mapa de las biorregiones para cada uno de los años de la serie. Claramente observamos, tal y como muestra el modelo, que la diversidad es mayor en las zonas de aguas mas salinas. Siempre puede encontrarse alguna excepción y debemos recordar que se trata de una influencia parcial. No habrá ninguna variable que permita explicar el 100% de la variabilidad de la diversidad.

El análisis de SIMPER (porcentajes de similitud) realizado para cuantificar la contribución de las especies en las biorregiones observadas calcula el porcentaje de diferencia en la composición por especies de dos o mas grupos pre-establecidos. Si dicho porcentaje supera el 70% podemos suponer que los grupos son diferentes. En el caso de las biorregiones basadas en la salinidad de la capa de mezcla, la diferencia entre grupos encontrada fue del 72.5%.

5. REFERENCIAS

Alvarez-Berastegui, D., Hidalgo, M., Tugores, M.P., Reglero, P., Aparicio-González, A., Ciannelli, L., Juza, M., Mourre, B., Pascual, A., López-Jurado, J.L., García, A., Rodríguez, J.M., Tintoré, J., Alemany, F. (2016). Pelagic seascape ecology for operational fisheries oceanography: Modelling and predicting spawning distribution of Atlantic bluefin tuna in Western Mediterranean. *ICES Journal of Marine Science*, 73 (7), pp. 1851-1862. DOI: 10.1093/icesjms/fsw041

Bakun A (2006) Fronts and eddies as key structures in the habitat of marine fish larvae: opportunity, adaptive response and competitive advantage. *Scientia Marina* 70:105-122

Bakun, A.(2010). Linking climate to population variability in marine ecosystems characterized by non-simple dynamics: Conceptual templates and schematic constructs. *Journal of Marine Systems*, 79 (3-4), pp. 361-373. DOI: 10.1016/j.jmarsys.2008.12.008

Coll, M., Piroddi, C., Steenbeek, J., Kaschner, K., Ben RaisLasram, F., Aguzzi, J., ... Voultsiadou, E. (2010). The biodiversity of the Mediterranean Sea: Estimates, patterns, and threats. *PLoS ONE*, 5, e11842. <https://doi.org/10.1371/journal.pone.0011842>

Dickey-Collas, M., McQuatters-Gollop, A., Bresnan, E., Kraberg, A.C., Manderson, J.P., Nash, R.D.M., Otto, S.A., Sell, A.F., Tweddle, J.F., Trenkel, V.M. (2017). Pelagic habitat: Exploring the concept of good environmental status. *ICES Journal of Marine Science*, 74 (9), pp. 2333-2341. DOI: 10.1093/icesjms/fsx158

European Commission, 2017. Commission decision (EU) 2017/848 of 17 May 2017 laying down criteria and methodological standards on good environmental status of marine waters and specifications and standardised methods for monitoring and assessment, and repealing Decision 2010/477/EU. *Official Journal of the European Union* L, 125: 43–74.

Granger, V., Fromentin, J., Bez, N., Relini, G., Meynard, C.N., Gaertner, J., Maiorano, P., Garcia, C., Follesa, C., Gristina, M., Peristeraki, P., Brind, A., Carbonara, P., Charilaou, C., Esteban, A., Jadaud, A., Joksimovic, A., Kallianiotis, A., Kolitari, J., Manfredi, C., Massuti, E., Mifsud, R., Quetglas, A., Refes, W., Sbrana, M., Vrgoc, N., Teresa, M., Mérigot, B., (2015). Progress in Oceanography Large-scale spatio-temporal monitoring highlights hotspots of demersal fish diversity in the Mediterranean Sea. *Progress in Oceanography* 130, 65–74. <https://doi.org/10.1016/j.pocean.2014.10.002>

Hidalgo M, Reglero P, Álvarez-Berastegui D, Torres A, Álvarez I, Rodríguez J, Carbonell A, Balbín R, Alemany F (2015) Hidden persistence of salinity and productivity gradients shaping pelagic diversity in highly dynamic marine ecosystems. *Marine Environmental Research*, 104, pp. 47-50. DOI: 10.1016/j.marenvres.2015.01.001.

Miller BS, Kendall AW (2009) Early life history of marine fishes, Vol 36. University of California Press Berkeley

Myers, N., Mittermeier, R. A., Mittermeier, C. G., da Fonseca, G. A. B., & Kent, J. (2000). Biodiversity hotspots for conservation priorities. *Nature*, 403, 853–858. <https://doi.org/10.1038/35002501>

Tugores, M.P., Ordines, F., Guijarro, B., García-Ruiz, C., Esteban, A., Massutí, E. (2019). Essential fish habitats and hotspots of nekto-benthic diversity and density in the western Mediterranean. *Aquatic Conservation: Marine and Freshwater Ecosystems*, 29 (3), pp. 461-471. DOI: 10.1002/aqc.3031

UNEP-MAP RAC/SPA (2010). The Mediterranean Sea Biodiversity: state of the ecosystems, pressures, impacts and future priorities. By Bazairi, H., Ben Haj, S., Boero, F., Cebrian, D., De Juan, S., Limam, A., Lleonart, J., Torchia, G., and Rais, C., Ed. RAC/SPA, Tunis; 100 pages.



ANEXO

SCRIPT Y VISUALIZADOR DE HERRAMIENTAS

Se presentan en las siguientes páginas los scripts:

1. Abu_data.R.....Pag. 27
2. Biodiversity.R.....Pag. 34
3. myCTD.R.....Pag. 46
4. DIV+CTD.R.....Pag. 58
5. HOM_DATA.R.....Pag. 68
6. DIV+CTD+DENS.R.....Pag. 72
7. SHANNONGam_200224.R:.....Pag. 78
8. DIV_MAPS.R: **VISOR DE DATOS**.....Pag. 140

Breve explicación de cada uno:

1. Abu_data.R

```
# Uso de datos de abundancia de ictio y el conjunto de datos de larvas de túnidos en Access
# En este script los datos de abundancias de ictio y larvas de túnidos están limpias
# Comprobación duplicados
# Corrección valores extraños
# Eliminación datos no útiles
# Fusionando abundancias de atún e ictio en el mismo dataframe --> "00.ictiotuna.xlsx"
# ARCHIVOS DATOS USADOS: 01.new_tun.xlsx; 01.new_ictio.xlsx;
01.ictiotuna.xlsx; 01.sp_no_presentes.xlsx
```

2. Biodiversity.R

```
# De datos de abundancia a datos de densidad * 100m3 --> "02.ictiotuna_dens.xlsx"
# De forma ancha a forma larga para hacer una matriz de densidad de especies# Species density matrix
# Biodiversity indexes: SHANNON, 1 - SIMPSON, 1/SIMPSON and NUMBER SPECIES
```



```
# "03.ictiotuna_div.xlsx"  
# "03.ictio_div.xlsx"  
# "03.tuna_div.xlsx"
```

3. myCTD.R

```
# Crear mis datos CTD con encuestas y variables de interés a partir de datos CTD  
# Function moveme to order columns  
# archive final --> 01.myCTD.xlsx
```

4. DIV+CTD.R

```
# Fusión de datos de biodiversidad y datos CTD y datos de operación (profundidad de la estación)  
# moveme function to reorder columns  
# ordenar filas  
# biodiversity complete data + CTD --> "04.divCTD.xlsx"  
# ictio biodiverstiyy data + CTD --> "04.divictioCTD.xlsx"  
# tuna biodiversity data + CTD --> "04.divtunaCTD.xlsx"  
# diversity indexes for the three datasets + CTD --> "04.DIV_CTD.xlsx"
```

5. HOM_DATA.R

```
# Retirar la estación de muestreo más occidental para evitar el Cap de la Nao  
# Eliminar muestras nocturnas  
# Eliminar max_depth outliers  
# Eliminar estaciones duplicadas  
# Archivo final con datos homogéneos para análisis GAM --> 05.Hom_DATA.xlsx  
# Statistics summary of homogeneous data --> 05.Hom_DATA_summary.xlsx
```

6. DIV+CTD+DENS.R

```
# Calculo densidad larvaria * 100 m3 de cada especie en cada muestra  
# Unir DIV + CTD dataset con DENS dataset
```

Explorando proporciones de especies por muestras y años --> PhylaByYear2.xlsx

Archivo final --> 06.DIV_CTD_DENS.xlsx

7. **SHANNONgam_200224.R:**

ANALISIS GAM PARA DATOS HOMOGENOS usar los datos de
05.Hom_DATA.xlsx

8. **DIV_MAPS.R:** usar los datos de 05.Hom_DATA.xlsx

VISOR Mapas índices de diversidad

1. Abu_data.R

```
# ---- DIVERPEL ABUNDANCE DATA ----  
# Author: ESTHER BARBER  
# Created on: NOV 2019  
# Modified: JAN 2020  
#  


---



---

  
# Using abundance data from ictio and tuna dataset in access  
# In this script abundance data of tuna and ictio are cleaned.  
# Checking for duplicates  
# Correcting strange values  
# Removing non useful variables  
# Merging tuna and ictio abundances in the same data frame --> "00.ictiotuna.xlsx"  
# FILES: 01.new_tun.xlsx; 01.new_ictio.xlsx; 01.ictiotuna.xlsx; 01.sp_no_presentes.xlsx  
#  


---



---

  
# ---- PACKAGES ----  
library(tidyverse)  
library(xlsx)  
library(dplyr)  
library(readxl)  
# ----  
  
# ---- FUNCTIONS ----  
# ---- Remove zero cols ----  
remove_zero_cols <- function(df) {  
  rem_vec <- NULL  
  for(i in 1:ncol(df)){  
    this_sum <- summary(df[,i])  
    zero_test <- length(which(this_sum == 0))  
    if(zero_test == 6) {  
      rem_vec[i] <- names(df)[i]  
    }  
  }  
  features_to_remove <- rem_vec[!is.na(rem_vec)]  
  rem_ind <- which(names(df) %in% features_to_remove)  
  df <- df[,-rem_ind]  
  return(df)  
}
```

#

---- DIVERPEL TUNA ABUNDANCE ----

---- TUNA DATA ----

```
# Dataset tunidos 00_20190704_abs_tr_tunidos.xlsx
tun <-
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/00_201
90704_abs_tr_tunidos.xlsx")
tun <- data.frame(tun)
# 653 observaciones con 22 variables
str(tun)
rownames(tun)
colnames(tun)
names(tun)
```

---- DUPLICADOS: comprobar que hay un unico colector de bongo en cada una de las muestras ----

```
# buscando filas duplicadas
duplicated(tun)
# comprobar el numero de observaciones por operacion
table(tun$to_idoperacion_num)
# todas las operaciones tienen una unica observacion (colector)
# comprobar que la suma de observaciones por colector coincida con el total de
operaciones
table(tun$tc_colectorn)
# 1 2
# 278 375
# Numero de estaciones muestreadas
length(unique(tun$tc_codestacion)) # 251
# no todas las estaciones se han muestreado en todas las campañas.
table(tun$tc_codestacion)
#
```

---- COLUMNAS CON TODO 0: Identificar si hay alguna columna con todo 0 ----

Significaria que esa especie fue identificada en años anteriores pero no en el periodo
2012-2016

```
tun_presentes <- colSums(tun[, 16:22])
```

```
# Thunnus atlanticus tiene abundancia 0
```

```
# Igual que en los datos de ictio, elimino la variable LSV.tunidos.NI
tun <- tun[, -19]
```

```
# Crear un dataset sin las especies que tienen abundancia 0
```

```
remove_zero_cols <- function(df) {
  rem_vec <- NULL
  for(i in 1:ncol(df)){
    this_sum <- summary(df[,i])
    zero_test <- length(which(this_sum == 0))
    if(zero_test == 6) {
      rem_vec[i] <- names(df)[i]
    }
  }
  features_to_remove <- rem_vec[!is.na(rem_vec)]
  rem_ind <- which(names(df) %in% features_to_remove)
  df <- df[,-rem_ind]
  return(df)
}
```

```
# ---- CLEANED TUNA DATA ----
```

```
# Nuevo dataset sin las columnas que son todo 0
new.tun <- remove_zero_cols(tun)
write.xlsx(new.tun, file="01.new_tun.xlsx")
# ----
```

```
# ---- DIVERPEL ICTIO ABUNDANCE ----
```

```
# ---- ICTIO DATA ----
```

```
# Dataset ictio 00_20190724_abs_tr_ictio.xlsx
ictio <-
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/00_201
90724_abs_tr_ictio.xlsx")
ictio <- data.frame(ictio)
# hay un total de 651 observaciones (dos menos que en tunidos)
names(ictio)
table(ictio$grupo_para_procesar) # ictio sin tunidos
```

```
# ---- DUPLICADOS: comprobar que hay un unico colector de bongo en cada una de las
muestras ----
```

```

# buscando filas duplicadas
duplicated(ictio)
x <- ictio[duplicated(ictio),] # data frame with 0 observations of 242 variables
# Significa que no hay duplicados
# comprobar el numero de observaciones por operacion
table(ictio$to_idoperacion_num)
# todas las operaciones tienen una unica observacion (colector)
# comprobar que la suma de observaciones por colector coincida con el total de
operaciones
table(ictio$tc_colectorn)
# 1 2
# 276 375
# Numero de estaciones muestreadas
length(unique(ictio$tc_codestacion)) # 250 hay una estacion menos que en tunidos
# no todas las estaciones se han muestreado en todas las campañas.
table(ictio$tc_codestacion)

# ---- DATOS RAROS: Elimino variables que no me interesan y datos raros (mail Itzi) ----
# Eliminar la taxa Myctophidae juvenil
ictio$Myctophidae.juvenil <- NULL # dim 651 obs x 241 variables
# Eliminar columna Apoda
ictio$Apoda <- NULL # dim 651 obs x 240 variables
# Eliminar columna Huevos
ictio$Huevos <- NULL # dim 651 obs x 239 variables
# Eliminar columna Larvas.NI
ictio$Larvas.NI <- NULL # dim 651 obs x 238 variables
# Eliminar columna LSV.NI
ictio$LSV.NI <- NULL # dim 651 obs x 237 variables
# Eliminar columna Restos
ictio$Restos <- NULL # dim 651 obs x 236 variables

# Juntar "Symphodus.B" y "Symphodus.B.". Sumo ambas columnas en la columna
Symphodus.B y elimino Symphodus.B.
table(ictio$Symphodus.B) # 0 2
# 650 1
table(ictio$Symphodus.B.) # 0 12
# 650 1
ictio$Symphodus.B <- ictio$Symphodus.B + ictio$Symphodus.B.
table(ictio$Symphodus.B) # 0 2 12
# 649 1 1
ictio$Symphodus.B. <- NULL # dim 651 obs x 235 variables

# Serranus.cabrilla tiene un valor de -999. Eliminarlo
ictio$Serranus.cabrilla [204] # [1] -999

```

```
has.neg <- apply(ictio[16:235], 1, function(row) any(row < 0))
which(has.neg) # [1] 204 row with the negative value
ictio$Serranus.cabrilla[204] <- 0
```

```
# ---- COLUMNAS TODO 0: Identificar si hay alguna columna con todo 0 ----
# Significaria que esa especie fue identificada en a?os anteriores pero no en el periodo
```

```
2012-2016
```

```
ictio_presentes <- colSums(ictio[, 16:235])
false.ictio <- ictio_presentes[ictio_presentes == 0 ]
# hay 90 especies que presentan abundancia 0 para los a?os seleccionados
write.xlsx(false.ictio, file="01.sp_no_presentes.xlsx")
```

```
# ---- CLEANED ICTIO DATA ----
```

```
# Nuevo dataset
```

```
new.ictio <- remove_zero_cols(ictio) # 650 obs x 145 variables
write.xlsx(new.ictio, file="01.new_ictio.xlsx")
```

```
# ----
```

```
# ---- DIVERPEL COMPLETE ABUNDANCE DATA ----
```

```
# ---- CHECKING ROW DIFFERENCES BETWEEN TUNA AND ICTIO ----
```

```
# Comparando los datos de tunidos e ictio
```

```
table(new.ictio$tc_campagna)
```

```
table(new.tun$tc_campagna)
```

```
# el numero de observaciones de la campagna BF0615 no coincide entre ictio y tunidos
```

```
# Creo un dataframe de la campagna BF0615
```

```
ictio15 <- new.ictio[new.ictio$tc_campagna == "BF0615", ]
```

```
tun15 <- new.tun[new.tun$tc_campagna == "BF0615", ]
```

```
# Extraigo el vector que corresponde a la variable operacion
```

```
ope_ictio15 <- ictio15$to_idoperacion_num # length 92
```

```
ope_tun15 <- tun15$to_idoperacion_num # length 94
```

```
# comparo ambos vectores
```

```
setdiff(ope_tun15, ope_ictio15) # [1] 4088 4091 operaciones presentes en tunidos pero no en ictio
```

```
tun15[tun15$to_idoperacion_num == "4088", ] # abundancia de tunidos es 0
```

```
tun15[tun15$to_idoperacion_num == "4091", ] # abundancia de tunidos es 0
```

```
# estas dos observaciones no nos interesan. Quedaran eliminadas cuando fusionemos el conjunto de datos de ictio y tunidos
```

```

# las operaciones que no aparecen en ictio perteneces a las estaciones
st_ictio15 <- ictio15$tc_codestacion # 92
st_tun15 <- tun15$tc_codestacion # 94
# comparo ambos vectores
setdiff(st_tun15, st_ictio15) # [1] "886" "889"
# ---- RENAME & REMOVE VARIBALES ----
# renombrar y eliminar variables que no aportan nada a la clasificaicion y diferenciacion.
# Transformar las otras (que no son sps) a clase factor
tuna <- new.tun
tuna <- tuna %>%
  dplyr::select(-tc_flag_procesado, -to_tipo_estacion, -tpt_larval_index, -tc_estructura, -
  tp_tipo_pesca, -tc_cont_estructura,
  -tc_colectorn, -grupo_para_procesar, -LSV.tunidos.NI) %>%
  mutate_at(vars(tc_id_colector_autonum,to_idoperacion_num,tc_campania,tc_codestacion,tc_orden_bd), funs(factor))
# 653 obs x 12 variables

ictio <- new.ictio
ictio <- ictio %>%
  dplyr::select(-tc_flag_procesado, -to_tipo_estacion, -tpt_larval_index, -tc_estructura, -
  tp_tipo_pesca, -tc_cont_estructura,
  -tc_colectorn, -grupo_para_procesar) %>%
  mutate_at(vars(tc_id_colector_autonum,to_idoperacion_num,tc_campania,tc_codestacion,tc_orden_bd), funs(factor))
# 651 obs x 137 variables

# ---- IGUALAR DATASETS: Extraigo un vector de operaciones comunes a las dos tablas de datos y lo ordeno de menor a mayor ----
comun <- factor(sort(intersect(tuna$"to_idoperacion_num",
  ictio$"to_idoperacion_num")))
# hay 651 operaciones comunes = tengo datos abundancia de ictio y de tunidos

# ---- MERGING DATAFRAMES: Las variables que use para fusionar tiene que tener datos en comun ----
# FUSIONAR ICTIO + TUN: fusionar los dos conjuntos de datos basandonos en el numero de operacion
# como hemos visto, quedaran fuera dos operaciones (4088 y 4091) ya que estas no estan presentes
# en ictio.
# Comprobar que los niveles de las variables factor que use para fusionar tengan los mismos nombres

```

levels(ictio\$tc_campania)

levels(tuna\$tc_campania)

```
ictiotuna <- merge(ictio, tuna, by=c("tc_campania", "to_idoperacion_num",
"tc_id_colector_autonum", "tc_codestacion", "tc_orden_bd",
"tp_HoraInicoGMT", "tc_m3" ))
```

```
# ---- COMPLETE DATASET ICTIO + TUNA ----
```

```
# data.frame con 651 obs x 142 variables ((137 ictio + 12 tuna) - 7 merging variables)
```

```
write.xlsx(ictiotuna, file="01.ictiotuna.xlsx")
```

```
# ----
```

```
# ---- [[[[[[[[[[[[[ THE END ]]]]]]]]]]]]] ----
```

2.Biodiversity.R

---- DIVERPEL BIODIVERSITY ----

Author: ESTHER BARBER

Create on: NOV 2019

Modified: JAN 2020

From abundance data to density * 100m3 data --> "02.ictiotuna_dens.xlsx"

From wide shape to long shape to make a species density matrix

Species density matrix

Biodiversity indexes: SHANNON, 1 - SIMPSON, 1/SIMPSON and NUMBER SPECIES

"03.ictiotuna_div.xlsx"

"03.ictio_div.xlsx"

"03.tuna_div.xlsx"

#

---- PACKAGES ----

library(tidyverse)

library(xlsx)

library(dplyr)

library(readxl)

library(reshape2)

library(vegan) # para calcular indices de biodiversidad

---- DIVERPEL ABUNDANCES DATA ----

DATA



Dataset tunidos 00.ictiotuna.xlsx

```
data
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/01.ictiotuna.xlsx")
```

```
data <- data.frame(data)
```

```
str(data)
```

```
data <- data %>%
```

```
mutate_at(vars(tc_campania,to_idoperacion_num,tc_id_colector_autonum,tc_codestacion,tc_orden_bd), funs(factor))
```

```
# 651 obs x 142 variables
```

```
#
```

```
# ---- DIVERPEL LARVAE DENSITY DATA ----
```

```
# DENSIDAD: Transformar los datos de abundancia a datos de densidad de larvas en 100m3 de cada especie en cada operacion
```

```
# Trabajamos con densidades porque no todos los muestreos filtraron el mismo volumen de agua (tc_m3)
```

```
# Asi normalizamos los datos y permitimos hacer comparaciones
```

```
# abundance sps i/volume i * 100
```

```
data_dens <- with(data,
cbind(tc_campania,to_idoperacion_num,tc_id_colector_autonum,tc_codestacion,tc_orden_bd,tp_HoraInicoGMT,
```

```
tc_m3,
```

```
data[!names(data) %in%
```

```
c("tc_campania","to_idoperacion_num","tc_id_colector_autonum","tc_codestacion","tc_orden_bd",
```

```
"tp_HoraInicoGMT","tc_m3")]/tc_m3*100))
```

```
# 651 obs x 142 variables
```

```
write.xlsx(data_dens, file="02.ictiotuna_dens.xlsx")
```

#

---- COMPLETE DIVERSITY ----

FROM WIDE SHAPE TO LONG SHAPE: Reordeno el conjunto de datos

de manera que los nombres de las especies queden recogidos en una misma variable y la densidad de las mismas, en otra.

dens_long <- melt(data_dens, id=c("tc_campagna", "to_idoperacion_num", "tc_id_colector_autonomo", "tc_codestacion", "tc_orden_bd",

"tp_HoraInicoGMT", "tc_m3"), variable.name = "Phyla",
value.name = "Density")

dens_long --> 87885 x 9 variables (142(vbles data_dens) - 135(sp) + 2(vbles(vble & value)))

names(dens_long)

str(dens_long)

#

---- MATRIZ DE ESPECIES ----

MATRIZ: Necesito crear una matriz de densidad de especies para los calculos de biodiversidad

STT <- as.factor(sort(unique(dens_long\$to_idoperacion_num)))

uso la variable to_idoperacion_num para no perder el rastro de los datos del CTD y porque las estaciones se repiten en los distintos años

SSP <- as.vector(sort(unique(dens_long\$Phyla)))

crea un vector ordenado con los valores únicos de la variable nombre de especie (Phyla)

Crear una matriz con 0's en la que las especies son las filas y las columnas son las operaciones

```
m_dens <- matrix(0, nrow = length(SSP), ncol = length(STT), byrow = T, dimnames = list(SSP, STT))
```

```
# lleno la matriz de datos
```

```
for(i in 1:nrow(dens_long))
```

```
{ m_dens[rownames(m_dens)==dens_long[i,8],colnames(m_dens)==dens_long[i,2]] <-
dens_long[i,9]}
```

```
# rownames columna 8 nombre de las especies en la columna variable
```

```
# colnames columna 2 numero de operacion "tc_id_operacion_num"
```

```
# datos de la matriz columna 9 densidad de larvas de tunidos
```

```
tm_dens <- t(m_dens)
```

```
# la matriz m tiene un unico dato de densidad de cada una de las especies en cada estacion
```

```
# traspone la matriz de manera que las sps son las columnas y las estaciones son las filas
```

```
#
```

```
# ---- CALCULOS DIVERSIDAD ----
```

```
# Crea un data frame vacio con todas las variables
```

```
# STT es el numero de operacion
```

```
# Tambien creare dos nuevas variables
```

```
# Shannon = biodiversidad especifica diversity()
```

```
# SpsNum = numero de especies specnumber()
```

```
# 1-Simposn = indice de simpson (1-D) <- diversity(data, "simpson")
```

```
# inv_Simps = inversa de simposon (1/D) <- diversity(data, "inv")
```

```
ictiotuna_div <- data.frame(orden=STT, estacion= rep(NA, length(STT)),
```

```
survey= rep(NA, length(STT)), horaGMT= rep(NA, length(STT)), colector=
rep(NA, length(STT)),
```

```
orden_bd=rep(NA, length(STT)), Volumen=rep(NA, length(STT)),
```

$\text{Shannon} = \text{rep}(\text{NA}, \text{length}(\text{STT}))$, $\text{SpsNum} = \text{rep}(\text{NA}, \text{length}(\text{STT}))$,
 $"1\text{-Simpson} = \text{rep}(\text{NA}, \text{length}(\text{STT}))$, $\text{inv_Simps} = \text{rep}(\text{NA}, \text{length}(\text{STT}))$)

```

for(z in 1:length(STT)) {

  #z <- 35 ?significa que tenemos 35 transectos? En nuestro caso serian 149

  station <- STT[z]

  ictiotuna_div[ictiotuna_div$orden == station, 2] <-
    dens_long$tc_codestacion[dens_long$to_idoperacion_num==station][1]

  ictiotuna_div[ictiotuna_div$orden == station, 3] <-
    dens_long$tc_campagna[dens_long$to_idoperacion_num==station][1]

  ictiotuna_div[ictiotuna_div$orden == station, 4] <-
    dens_long$tp_HoraInicoGMT[dens_long$to_idoperacion_num==station][1]

  ictiotuna_div[ictiotuna_div$orden == station, 5] <-
    dens_long$tc_id_colector_autonom[dens_long$to_idoperacion_num==station][1]

  ictiotuna_div[ictiotuna_div$orden == station, 6] <-
    dens_long$tc_orden_bd[dens_long$to_idoperacion_num==station][1]

  ictiotuna_div[ictiotuna_div$orden == station, 7] <-
    dens_long$tc_m3[dens_long$to_idoperacion_num==station][1]

  ictiotuna_div[ictiotuna_div$orden == station, 8] <- diversity(tm_dens)[z]

  ictiotuna_div[ictiotuna_div$orden == station, 9] <- specnumber(tm_dens)[z]

  ictiotuna_div[ictiotuna_div$orden == station, 10] <- diversity(tm_dens, "simpson")[z]

  ictiotuna_div[ictiotuna_div$orden == station, 11] <- diversity(tm_dens, "inv")[z]
}

  
```

```

head(ictiotuna_div)
str(ictiotuna_div)
table(ictiotuna_div$survey)

  ictiotuna_div$survey <- factor(ictiotuna_div$survey, levels=c("1","2","3","4","5"),
  labels=c("ATAME0612", "BF0613","BF0614","BF0615","BF0616"))
  
```

```
write.xlsx(ictiotuna_div, file="03.ictiotuna_div.xlsx")
```

```
#
```

```
# ----
```

```
# ---- ICTIO DIVERSITY ----
```

```
# DATA: Ictio species in columns 8:137
```

```
ictio_dens <- data_dens[, 1:137]
```

```
# FROM WIDE SHAPE TO LONG SHAPE: Reordeno el conjunto de datos.
```

```
# de manera que los nombres de las especies queden recogidos en una misma variable y la densidad de las mismas, en otra.
```

```
ictio_dens_long <- melt(ictio_dens, id=c("tc_campagna", "to_idoperacion_num", "tc_id_colector_autonom", "tc_codestacion", "tc_orden_bd",
```

```
"tp_HoraInicoGMT","tc_m3"), variable.name = "Phyla", value.name = "Density")
```

```
# ictio_dens_long --> 84630 x 9 variables (137(vbles ictio_dens) - 130(sp) + 2(vbles(vble & value)))
```

```
names(ictio_dens_long)
```

```
#
```

```
# ---- MATRIZ DE ESPECIES ----
```

```
# MATRIZ: Necesito crear una matriz de densidad de especies para los calculos de biodiversidad
```

```
STT <- as.factor(sort(unique(ictio_dens_long$to_idoperacion_num)))
```

uso la variable to_idoperacion_num para no perder el rastro de los datos del CTD y porque las estaciones se repiten en los distintos años

```
SSP <- as.vector(sort(unique(ictio_dens_long$Phyla)))
```

```
# crea un vector ordenado con los valores únicos de la variable nombre de especie (Phyla)
```

```
# Crear una matriz con 0's en la que las especies son las filas y las columnas son las operaciones
```

```
m_ictio_dens <- matrix(0, nrow = length(SSP), ncol = length(STT), byrow = T, dimnames = list(SSP, STT))
```

```
# lleno la matriz de datos
```

```
for(i in 1:nrow(ictio_dens_long))
```

```
{m_ictio_dens[rownames(m_ictio_dens)==ictio_dens_long[i,8],colnames(m_ictio_dens)==ictio_dens_long[i,2]] <- ictio_dens_long[i,9]}
```

```
# rownames columna 8 nombre de las especies en la columna variable
```

```
# colnames columna 2 numero de operación "tc_id_operacion_num"
```

```
# datos de la matriz columna 9 densidad de larvas de tunidos
```

```
tm_ictio_dens <- t(m_ictio_dens)
```

```
# la matriz m tiene un único dato de densidad de cada una de las especies en cada estación
```

```
# traspone la matriz de manera que las sps son las columnas y las estaciones son las filas
```

```
#
```

```
# ---- CALCULOS DIVERSIDAD ----
```

```
# Crea un data frame vacío con todas las variables
```

```
# STT es el número de operación
```

```
# También creará dos nuevas variables
```

```
# Shannon = biodiversidad específica diversity()
```

```
# SpsNum = número de especies specnumber()
```

```
# 1-Simpson = índice de Simpson (1-D) <- diversity(data, "simpson")
```

inv_Simps = inversa de simposon (1/D) <- diversity(data, "inv")

```

ictio_div <- data.frame(orden=STT, estacion= rep(NA, length(STT)),
                        survey= rep(NA, length(STT)), horaGMT= rep(NA, length(STT)), colector=
                        rep(NA, length(STT)),
                        orden_bd=rep(NA, length(STT)), Volumen=rep(NA, length(STT)),
                        Shannon=rep(NA, length(STT)), SpsNum=rep(NA, length(STT)),
                        "1-Simpson"=rep(NA, length(STT)), inv_Simps=rep(NA, length(STT)))

for(z in 1:length(STT)) {

  #z <- 35 ?significa que tenemos 35 transectos? En nuestro caso serian 149

  station <- STT[z]

  ictio_div[ictio_div$orden == station, 2] <-
    ictio_dens_long$tc_codenstacion[ictio_dens_long$to_idoperacion_num==station][1]

  ictio_div[ictio_div$orden == station, 3] <-
    ictio_dens_long$tc_campagna[ictio_dens_long$to_idoperacion_num==station][1]

  ictio_div[ictio_div$orden == station, 4] <-
    ictio_dens_long$tp_HoraInicoGMT[ictio_dens_long$to_idoperacion_num==station][1]

  ictio_div[ictio_div$orden == station, 5] <-
    ictio_dens_long$tc_id_colector_autonom[ictio_dens_long$to_idoperacion_num==station][1]

  ictio_div[ictio_div$orden == station, 6] <-
    ictio_dens_long$tc_orden_bd[ictio_dens_long$to_idoperacion_num==station][1]

  ictio_div[ictio_div$orden == station, 7] <-
    ictio_dens_long$tc_m3[ictio_dens_long$to_idoperacion_num==station][1]

  ictio_div[ictio_div$orden == station, 8] <- diversity(tm_ictio_dens)[z]

  ictio_div[ictio_div$orden == station, 9] <- specnumber(tm_ictio_dens)[z]

  ictio_div[ictio_div$orden == station, 10] <- diversity(tm_ictio_dens, "simpson")[z]

  ictio_div[ictio_div$orden == station, 11] <- diversity(tm_ictio_dens, "inv")[z]

}

```

```
head(ictio_div)
```

```
ictio_div$survey      <-      factor(ictio_div$survey,      levels=c("1","2","3","4","5"),
labels=c("ATAME0612", "BF0613","BF0614","BF0615","BF0616"))
```

```
str(ictio_div)
```

```
table(ictio_div$survey)
```

```
write.xlsx(ictio_div, file="03.ictio_div.xlsx")
```

```
#
```

```
# ----
```

```
# ---- TUNA DIVERSITY ----
```

```
# DATA: Tuna species in columns 138:142
```

```
tuna_dens <- data_dens[, -c(8:137)]
```

```
# FROM WIDE SHAPE TO LONG SHAPE: Reordeno el conjunto de datos.
```

```
# de manera que los nombres de las especies queden recogidos en una misma variable y la densidad de las mismas, en otra.
```

```
tuna_dens_long      <-      melt(tuna_dens,      id=c("tc_campagna",      "to_idoperacion_num",
"tc_id_colector_autonom", "tc_codestacion", "tc_orden_bd",
```

```
                      "tp_HoraInicoGMT","tc_m3"), variable.name = "Phyla", value.name
= "Density")
```

```
# tuna_dens_long --> 3255 x 9 variables (12(vbles tuna_dens) - 5(sp) + 2(vbles(vble & value)))
```

```
names(tuna_dens_long)
```

#

---- MATRIZ DE ESPECIES ----

MATRIZ: Necesito crear una matriz de densidad de especies para los calculos de biodiversidad

```
STT <- as.factor(sort(unique(tuna_dens_long$to_idoperacion_num)))
```

uso la variable to_idoperacion_num para no perder el rastro de los datos del CTD y porque las estaciones se repiten en los distintos años

```
SSP <- as.vector(sort(unique(tuna_dens_long$Phyla)))
```

crea un vector ordenado con los valores únicos de la variable nombre de especie (Phyla)

Crear una matriz con 0's en la que las especies son las filas y las columnas son las operaciones

```
m_tuna_dens <- matrix(0, nrow = length(SSP), ncol = length(STT), byrow = T, dimnames = list(SSP, STT))
```

lleno la matriz de datos

```
for(i in 1:nrow(tuna_dens_long))
```

```
{m_tuna_dens[rownames(m_tuna_dens)==tuna_dens_long[i,8],colnames(m_tuna_dens)==tuna_dens_long[i,2]] <- tuna_dens_long[i,9]}
```

rownames columna 8 nombre de las especies en la columna variable

colnames columna 2 numero de operación "tc_id_operacion_num"

datos de la matriz columna 9 densidad de larvas de tunidos

```
tm_tuna_dens <- t(m_tuna_dens)
```

la matriz m tiene un único dato de densidad de cada una de las especies en cada estación

traspone la matriz de manera que las sps son las columnas y las estaciones son las filas

#

---- CALCULOS DIVERSIDAD ----

```
# Crea un data frame vacio con todas las variables
# STT es el numero de operacion
# Tambien creare dos nuevas variables
# Shannon = biodiversidad especifica diversity()
# SpsNum = numero de especies specnumber()
# 1-Simposn = indice de simpson (1-D) <- diversity(data, "simpson")
# inv_Simps = inversa de simposon (1/D) <- diversity(data, "inv")
```

```
tuna_div <- data.frame(orden=STT, estacion= rep(NA, length(STT)),
survey= rep(NA, length(STT)), horaGMT= rep(NA, length(STT)), colector=
rep(NA, length(STT)),
orden_bd=rep(NA, length(STT)), Volumen=rep(NA, length(STT)),
Shannon=rep(NA, length(STT)), SpsNum=rep(NA, length(STT)),
"1-Simpson"=rep(NA, length(STT)), inv_Simps=rep(NA, length(STT)))
```

```
for(z in 1:length(STT)) {
#z <- 35 ?significa que tenemos 35 transectos? En nuestro caso serian 149
station <- STT[z]
tuna_div[tuna_div$orden == station, 2] <-
tuna_dens_long$tc_codestacion[tuna_dens_long$to_idoperacion_num==station][1]
tuna_div[tuna_div$orden == station, 3] <-
tuna_dens_long$tc_campagna[tuna_dens_long$to_idoperacion_num==station][1]
tuna_div[tuna_div$orden == station, 4] <-
tuna_dens_long$tp_HoraInicoGMT[tuna_dens_long$to_idoperacion_num==station][1]
tuna_div[tuna_div$orden == station, 5] <-
tuna_dens_long$tc_id_colector_autonum[tuna_dens_long$to_idoperacion_num==station][1]
tuna_div[tuna_div$orden == station, 6] <-
tuna_dens_long$tc_orden_bd[tuna_dens_long$to_idoperacion_num==station][1]
```

```
tuna_div[tuna_div$orden == station, 7] <-  
tuna_dens_long$tc_m3[tuna_dens_long$to_idoperacion_num==station][1]
```

```
tuna_div[tuna_div$orden == station, 8] <- diversity(tm_tuna_dens)[z]
```

```
tuna_div[tuna_div$orden == station, 9] <- specnumber(tm_tuna_dens)[z]
```

```
tuna_div[tuna_div$orden == station, 10] <- diversity(tm_tuna_dens, "simpson")[z]
```

```
tuna_div[tuna_div$orden == station, 11] <- diversity(tm_tuna_dens, "inv")[z]
```

```
}
```

```
head(tuna_div)
```

```
tuna_div$survey <- factor(tuna_div$survey, levels=c("1","2","3","4","5"),  
labels=c("ATAME0612", "BF0613", "BF0614", "BF0615", "BF0616"))
```

```
str(tuna_div)
```

```
table(tuna_div$survey)
```

```
write.xlsx(tuna_div, file="03.tuna_div.xlsx")
```

```
#
```

```
# ----
```

```
# ---- [[[[[[[[[[[ THE END ]]]]]]]]]]] ----
```

3. myCTD.R

```
# ---- CTD DATA ----
```

```
# Author: ESTHER BARBER
```

```
# Created on: NOV 2019
```

```
# Modified: JAN 2020
```

```
# Create my CTD data with surveys and variables of interest from CTD data from Dani
```

```
# Function moveme to order columns
```

```
# Final file --> 01.myCTD.xlsx
```

```
# _____
```

```
# ---- PACKAGES ----
```

```
library(tidyverse)
```

```
library(xlsx)
```

```
library(dplyr)
```

```
library(readxl)
```

```
library(naniar) # to plot NA's
```

```
library(ggplot2)
```

```
# ---- FUNCTIONS ----
```

```
# ---- Reordenar columnas del dataframe ----
```

```
moveme <- function (invec, movecommand) {
```

```
  movecommand <- lapply(strsplit(strsplit(movecommand, ";")[[1]],  

    ",|\s+"), function(x) x[x != ""])
```

```
  movelist <- lapply(movecommand, function(x) {
```

```
    Where <- x[which(x %in% c("before", "after", "first",  

      "last")):length(x)]
```

```
ToMove <- setdiff(x, Where)
```

```
list(ToMove, Where)
```

```
)
```

```
myVec <- invec
```

```
for (i in seq_along(movelist)) {
```

```
temp <- setdiff(myVec, movelist[[i]][[1]])
```

```
A <- movelist[[i]][[2]][1]
```

```
if (A %in% c("before", "after")) {
```

```
ba <- movelist[[i]][[2]][2]
```

```
if (A == "before") {
```

```
after <- match(ba, temp) - 1
```

```
}
```

```
else if (A == "after") {
```

```
after <- match(ba, temp)
```

```
}
```

```
}
```

```
else if (A == "first") {
```

```
after <- 0
```

```
}
```

```
else if (A == "last") {
```

```
after <- length(myVec)
```

```
}
```

```
myVec <- append(temp, values = movelist[[i]][[1]], after = after)
```

```
}
```

```
myVec
```

```
}
```

```
# ----
```

---- DATA ----

Datos CTD 20191120_db_esther_dan.txt

CTD

read.delim("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/Base de datos/20191122_db_esther.txt")
<-

dim 968 obs x 68 vbls

names(CTD)

str(CTD)

Surveys es un factor con 11 niveles

a mi solo me interesan 5 campañas (niveles)

table(CTD\$survey)

Elimino las campañas que no me interesan

CTD\$survey <- as.character(CTD\$survey) # cambiar clase de la variable de factor a carácter para poder retener las líneas (campañas) que me interesan

myCTD <- CTD[CTD\$survey %in% c("ATAME0612", "BF0613", "BF0614", "BF0615", "BF0616"),]

table(myCTD\$survey)

705 obs x 68 variables

Elimino todas las variables que no me interesan

names(myCTD)

myCTD <- myCTD %>%

dplyr::select(-id_netcolector, -st_order, -date, -gear, -resampled_st, -within_systemat_dates, -type_st,

-arochei_n, -ealletteratus_n, -kpelamis_n, -talalunga_n, -atlanticus_n, -thynnus_n, -pnoctiluca_n,

-volume, -number_larvae, -number_species, -spp_richness) %>%

mutate_at(vars(id_operation, survey, st_id), funs(factor)) # dim 705 obs x 50 vbls

names(myCTD)

#

Esto ocurría en el archivo de datos 20191120_db_esther_dan

Hay operaciones duplicadas 6 operaciones con 343 observaciones. Son del seguimiento de una boya lagrangiana

id_operation 3542, 3555, 3558, 3604, 3605 y 3614.

identificar el numero de observaciones distintas

nrow(distinct(myCTD\$id_operation)) # 705

Can choose to keep all other variables as well

myCTD <- distinct(myCTD, id_operation, .keep_all = TRUE) # 705 obs x 43 vbles

#

---- FINAL CTD DATA ----

Reordeno las columnas de la tabla de datos

names(myCTD)

moveme <- function (invec, movecommand) {

movecommand <- lapply(strsplit(strsplit(movecommand, ";")[[1]],

", "\\\\s+"), function(x) x[x != ""])

movelist <- lapply(movecommand, function(x) {

Where <- x[which(x %in% c("before", "after", "first",

"last")):length(x)]

ToMove <- setdiff(x, Where)

list(ToMove, Where)

)

myVec <- invec

for (i in seq_along(movelist)) {

temp <- setdiff(myVec, movelist[[i]][[1]])

```

A <- movelist[[i]][[2]][1]

if (A %in% c("before", "after")) {

  ba <- movelist[[i]][[2]][2]

  if (A == "before") {

    after <- match(ba, temp) - 1

  }

  else if (A == "after") {

    after <- match(ba, temp)

  }

}

else if (A == "first") {

  after <- 0

}

else if (A == "last") {

  after <- length(myVec)

}

myVec <- append(temp, values = movelist[[i]][[1]], after = after)

}

myVec

}

myCTD <- myCTD[moveme(names(myCTD), "tow_duration before min_depth")]

write.xlsx(myCTD, file="01.myCTD.xlsx")

# ----

# ---- EXPLORING CTD DATA ----

```

Analysis based on:
https://mareds.github.io/r_course/ex_case_study_1_solutions.html#2:_assessing_the_data_quality

---- DUPLICATED STATIONS ----

Number of OPERATIONS sampled per SURVEY â†’ balanced?

```
myCTD %>%
  dplyr::select(id_operation, year) %>%
  distinct() %>%
  group_by(year) %>%
  count()
```

Number of stations sampled per SURVEY â†’ balanced?

```
myCTD %>%
  dplyr::select(st_id, year) %>%
  distinct() %>%
  group_by(year) %>%
  count()
```

which stations were sampled more than once?

```
x <- myCTD %>%
  #group_by(year)%>%
  count(st_id) %>% filter(n > 1)
x <- data.frame(x)
write.xlsx(x, file="St_duplo.xlsx")
```

Sampling frequency per stations: how many stations are sampled more than once per SURVEY?

```
freq_stat <- myCTD %>%
  dplyr::select(st_id, id_operation, year) %>%
```

this code chunk sums up the number of samplings per station and year

```
group_by(year, st_id) %>%
  count() %>%
  # now we filter only stations repeatedly sampled per month
  filter(n > 1)
  # ----
```

---- MISSING VALUES ----

```
# Missing values
any(is.na(myCTD)) # [1] TRUE
sum(is.na(myCTD)) # [1] 7128 of 35250 cells (705x50)
```

plotting missing values displaying total NA% and NA% per variable for the whole data

```
CTD_nas <- vis_miss(myCTD, sort_miss=T) # sort.miss = T decresing % order
png(filename =
  paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/CTD_NAs.png"), width =
  35, height = 20, units = "cm", res = 400)
print(CTD_nas)
dev.off()
```

CTD_nas2 <- gg_miss_upset(myCTD, nsets = n_var_miss(myCTD))

```
png(filename =
  paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/CTD_NAs2.png"), width =
  30, height = 30, units = "cm", res = 400)
print(CTD_nas2)
dev.off()
```

% NA total data

% NA by year

```
p2 <- gg_miss_var(myCTD,
  facet = year, show_pct=T)

png(filename =
  paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/CTD_NAs_pct_by_yr.png"),
  width = 30, height = 30, units = "cm", res = 400)

print(p2)

dev.off()
```

exploring NAs distribution by cruise computing numbers

```
z <- myCTD %>%
  # we will group by cruise so we can count the sum per cruise
  group_by(year) %>%
  summarise(
    # we calculate the sum over the logical vectors for temp, sal, and doxy
    na_all = sum(is.na(tow_duration), is.na(min_depth), is.na(max_depth), is.na(mld),
      is.na(temp5), is.na(temp10), is.na(temp25), is.na(temp50), is.na(temp100),
      is.na(temp200), is.na(tmld),
      is.na(sal5), is.na(sal10), is.na(sal25), is.na(sal50), is.na(sal100), is.na(sal200),
      is.na(smld),
      is.na(ox5), is.na(ox10), is.na(ox25), is.na(ox50), is.na(ox100), is.na(ox200),
      is.na(omld),
      is.na(gvel_u), is.na(gvel_v), is.na(gvel),
      is.na(meanChl), is.na(chl_month), is.na(chl_week), is.na(vsi), is.na(tmldR)),
    # to compare with the overall number of measurement per cruise
    n_measure = n(),
    # calculate the proportion of NAs overall and per parameter
    na_prop = round( na_all / (n_measure*33), 33), # *33 as we have 33 parameters per measurement
    na_prop_tow_duration = round(sum(is.na(tow_duration)) / n_measure, 3),
```

```
na_prop_min_depth = round(sum(is.na(min_depth)) / n_measure, 3),  
na_prop_max_depth = round(sum(is.na(max_depth)) / n_measure, 3),  
na_prop_MLD = round(sum(is.na(mld)) / n_measure, 3),  
na_prop_Tem5 = round(sum(is.na(temp5)) / n_measure, 3),  
na_prop_Tem10 = round(sum(is.na(temp10)) / n_measure, 3),  
na_prop_Tem25 = round(sum(is.na(temp25)) / n_measure, 3),  
na_prop_Tem50 = round(sum(is.na(temp50)) / n_measure, 3),  
na_prop_tem100 = round(sum(is.na(temp100)) / n_measure, 3),  
na_prop_Tem200 = round(sum(is.na(temp200)) / n_measure, 3),  
na_prop_TMEZCLA = round(sum(is.na(tmld)) / n_measure, 3),  
na_prop_Sal5 = round(sum(is.na(sal5)) / n_measure, 3),  
na_prop_Sal10 = round(sum(is.na(sal10)) / n_measure, 3),  
na_prop_Sal25 = round(sum(is.na(sal25)) / n_measure, 3),  
na_prop_Sal50 = round(sum(is.na(sal50)) / n_measure, 3),  
na_prop_Sal100 = round(sum(is.na(sal100)) / n_measure, 3),  
na_prop_Sal200 = round(sum(is.na(sal200)) / n_measure, 3),  
na_prop_SMEZCLA = round(sum(is.na(smlld)) / n_measure, 3),  
na_prop_OXI5 = round(sum(is.na(ox5)) / n_measure, 3),  
na_prop_OXI10 = round(sum(is.na(ox10)) / n_measure, 3),  
na_prop_OXI25 = round(sum(is.na(ox25)) / n_measure, 3),  
na_prop_OXI50 = round(sum(is.na(ox50)) / n_measure, 3),  
na_prop_OXI100 = round(sum(is.na(ox100)) / n_measure, 3),  
na_prop_OXI200 = round(sum(is.na(ox200)) / n_measure, 3),  
na_prop_OMEZCLA = round(sum(is.na(omld)) / n_measure, 3),  
na_prop_gvel_u = round(sum(is.na(gvel_u)) / n_measure, 3),  
na_prop_gvel_v = round(sum(is.na(gvel_v)) / n_measure, 3),  
na_prop_gvel = round(sum(is.na(gvel)) / n_measure, 3),  
na_prop_meanChl = round(sum(is.na(meanChl)) / n_measure, 3),  
na_prop_chl_month = round(sum(is.na(chl_month)) / n_measure, 3),
```

```

na_prop_chl_week = round(sum(is.na(chl_week)) / n_measure, 3),  

na_prop_vsi = round(sum(is.na(vsi)) / n_measure, 3),  

na_prop_tmldR = round(sum(is.na(tmldR)) / n_measure, 3)  

) %>%

```

```

# lets sort and select the 8 cruises with the highest NA proportion  

arrange(desc(na_prop)) %>%  

top_n(34, na_prop)

```

```
write.xlsx(z, file="CTD_prop_NA.xlsx")
```

```
# ----
```

```
# ---- TEMPERATURE PROFILES ----
```

```
temp <- myCTD[, c(1:6, 22:27)]
```

```

temp_long <- melt(temp, id=c("id_operation", "survey", "st_id", "lat", "lon", "year"),  

variable.name = "depth", value.name = "temperature")  

temp_long$depth <- factor(temp_long$depth, levels=c("temp5", "temp10", "temp25",  

"temp50", "temp100", "temp200"),  

labels=c("-5", "-10", "-25", "-50", "-100", "-200"))

```

```
temp_long1 <- temp_long[order(temp_long$id_operation), ]
```

```
temp_long1 <- temp_long1 %>%  

mutate(depth=factor(depth),  

depth=factor(depth, levels=rev(levels(depth))))
```

```
ggplot(temp_long1, aes(x = temperature, y = depth)) +  

geom_point(colour = "orange2", alpha = .2) +

```

```
facet_wrap(~year) +  
  
theme_bw()  
  
# ----  
  
# ---- SALINITY PROFILES ----  
  
sal <- myCTD[, c(1:6,29:34)]  
  
  
sal_long <- melt(sal, id=c("id_operation", "survey", "st_id", "lat", "lon", "year"),  
variable.name = "depth", value.name = "salinity")  
  
sal_long$depth <- factor(sal_long$depth, levels=c("sal5", "sal10", "sal25", "sal50", "sal100",  
"sal200"),  
labels=c("-5", "-10", "-25", "-50", "-100", "-200"))  
  
  
sal_long1 <- sal_long[order(sal_long$id_operation), ]  
sal_long1 <- sal_long1 %>%  
mutate(depth=factor(depth),  
depth=factor(depth, levels=rev(levels(depth))))  
  
  
ggplot(sal_long1, aes(x=salinity, y=depth)) +  
geom_point(colour = "maroon4", alpha = .2) +  
facet_wrap(~year) +  
theme_bw()  
  
# ----  
  
# ---- OXYGEN PROFILES ----  
  
oxy <- myCTD[, c(1:7,36:41)]  
  
  
oxy_long <- melt(oxy, id=c("id_operation", "survey", "st_id", "lat", "lon", "year"),
```



Unión Europea
Fondo Europeo Marítimo y
de Pesca (FEMP)



```
variable.name = "depth", value.name = "oxygen")
```

```
oxy_long$depth <- factor(oxy_long$depth, levels=c("ox5", "ox10", "ox25", "ox50", "ox100", "ox200"),
```

```
labels=c("-5", "-10", "-25", "-50", "-100", "-200"))
```

```
oxy_long1 <- oxy_long[order(oxy_long$id_operation), ]
```

```
oxy_long1 <- oxy_long1 %>%
```

```
mutate(depth=factor(depth),
```

```
depth=factor(depth, levels=rev(levels(depth))))
```

```
low_oxy <- oxy_long1[!(oxy_long1$oxygen >=0),, drop=T]
```

```
oxy_long1 <-
```

```
ggplot(oxy_long1, aes(x=oxygen, y=depth)) +
```

```
geom_point(colour = "cadetblue4", alpha = .2) +
```

```
facet_wrap(~year) +
```

theme_bw()

Hay NA's. Eliminarlos para poder hacer el grÁ;fico.

4. DIV+CTD.R

---- DIVERPEL: CTD + BIODIVERSITY ----

Author: ESTHER BARBER

Created on: JAN 2020

Merging biodiversity data and CTD data and operation data (station depth)

moveme function to reorder columns

arranging rows

biodiversity complete data + CTD --> "04.divCTD.xlsx"

ictio biodiverstiyy data + CTD --> "04.divictioCTD.xlsx"

tuna biodiversity data + CTD --> "04.divtunaCTD.xlsx"

diversity indexes for the three datasets + CTD --> "04.DIV_CTD.xlsx"

---- PACKAGES ----

library(tidyverse)

library(xlsx)

library(dplyr)

library(readxl)

---- FUNCTIONS ----

to arrange columns in the dataframe

moveme <- function (invec, movecommand) {

```
movecommand <- lapply(strsplit(strsplit(movecommand, ";")[[1]],  
",|\s+"), function(x) x[x != ""])
```

```
movelist <- lapply(movecommand, function(x) {
```

Where <- x[which(x %in% c("before", "after", "first",

"last")):length(x)]

```
ToMove <- setdiff(x, Where)
```

```
list(ToMove, Where)
```

```
)
```

```
myVec <- invec
```

```
for (i in seq_along(movelist)) {
```

```
temp <- setdiff(myVec, movelist[[i]][[1]])
```

```
A <- movelist[[i]][[2]][1]
```

```
if (A %in% c("before", "after")) {
```

```
ba <- movelist[[i]][[2]][2]
```

```
if (A == "before") {
```

```
after <- match(ba, temp) - 1
```

```
}
```

```
else if (A == "after") {
```

```
after <- match(ba, temp)
```

```
}
```

```
}
```

```
else if (A == "first") {
```

```
after <- 0
```

```
}
```

```
else if (A == "last") {
```

```
after <- length(myVec)
```

```
}
```

```
myVec <- append(temp, values = movelist[[i]][[1]], after = after)
```

```
)
```

```
myVec
```

```
}
```

```
# ----
```



---- CTD DATA ----

```
myCTD  
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/01.myC  
TD.xlsx")  
  
myCTD <- data.frame(myCTD)  
  
names(myCTD)  
  
str(myCTD)  
  
myCTD <- myCTD %>%  
  
  dplyr::select(-...1, -st_id) %>%  
  
  mutate_at(vars(id_operation, survey), funs(factor))
```

---- STATION DETPH DATA ----

```
zdata  
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/00.t_op  
eraciones.xlsx")  
  
zdata <- data.frame(zdata)  
  
names(zdata)  
  
str(zdata)  
  
zdata$to_idoperacion_num <- factor(zdata$to_idoperacion_num)  
  
zdata <- zdata[, c(1, 11)] # columna 1 es operaciÃ³n y columna 11 es prof. operacion  
  
# ----
```

---- DIVERSITY COMPLETE DATA ----

```
divdata  
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/03.ictiot  
una_div.xlsx")  
  
divdata <- data.frame(divdata)
```

names(divdata)

str(divdata)

divdata <- divdata %>%

dplyr::select(-...1, -horaGMT, -colector, -orden_bd) %>%

mutate_at(vars(orden, survey, estacion), funs(factor))

---- COMUN ELEMENTS BIODIVERSITY & CTD ----

comun <- factor(sort(intersect(divdata\$orden, myCTD\$id_operation))) # 650 operaciones en comun

setdiff(divdata\$estacion, myCTD\$st_id)

Las variables que use para fusionar tiene que tener datos en comun y ser factores

Los niveles de los factores comunes deben tener el mismo nombre

levels(divdata\$survey)

levels(myCTD\$survey)

Fusiono tabla de diversidad con tabla de CTD segun numero de operacion y campanya,

por estacion NO pq se repiten en las distintas campanyas

divCTD <- merge(myCTD, divdata, by.x=c("id_operation", "survey"),

by.y=c("orden", "survey")) # dim 650 obs x 55 vbles

---- COMUN ELEMENTS divCTD & ZDATA ----

comun1 <- factor(sort(intersect(divCTD\$id_operation, zdata\$to_idoperacion_num))) # 650 operaciones en comun

Fusiono tabla de diversidad con tabla de CTD segun numero de operacion y campana,

por estacion NO pq se repiten en las distintas campanas

```
divCTDz <- merge(zdata, divCTD, by.x=c("to_idoperacion_num"),
```

```
by.y=c("id_operation")) # dim 650 obs x 56 vbles
```

---- ARRANGING COLUMNS AND ROWS ----

```
head(divCTDz)
```

```
divCTDz <- divCTDz[moveme(names(divCTDz), "estacion before lat")]
```

```
divCTDz <- divCTDz[moveme(names(divCTDz), "to_profundidad_operacion after estacion")]
```

```
divCTDz <- divCTDz %>% arrange(year)
```

---- FINAL FILE ----

```
write.xlsx(divCTDz, file="04.divCTDz.xlsx")
```

---- ICTIO DIVERSITY DATA ----

```
divictio <-  
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/03.ictio  
_div.xlsx")
```

```
divictio <- data.frame(divictio)
```

```
names(divictio)
```

```
str(divictio)
```

```
divictio <- divictio %>%
```

```
dplyr::select(-...1, -horaGMT, -colector, -orden_bd) %>%
```

```
mutate_at(vars(orden, survey, estacion), funs(factor))
```

---- COMUN ELEMENTS DIVERSITY & CTD ----

```
comun <- factor(sort(intersect(divictio$orden, myCTD$id_operation))) # 650 operaciones en comun
```

```
# Las variables que use para fusionar tiene que tener datos en comun y ser factores
```

```
# Los niveles de los factores comunes deben tener el mismo nombre
```

```
levels(divictio$survey)
```

```
levels(myCTD$survey)
```

```
# Fusiono tabla de diversidad con tabla de CTD segun numero de operacion y campanya,
```

```
# por estacion NO pq se repiten en las distintas campanyas
```

```
divictioCTD <- merge(myCTD, divictio, by.x=c("id_operation", "survey"),
by.y=c("orden", "survey")) # dim 650 obs x 55 vbles
```

---- COMUN ELEMENTS divCTD & ZDATA ----

```
comun1 <- factor(sort(intersect(divictioCTD$id_operation, zdata$to_idoperacion_num))) # 650 operaciones en comun
```

```
# Fusiono tabla de diversidad con tabla de CTD segun numero de operacion y campanya,
```

```
# por estacion NO pq se repiten en las distintas campanyas
```

```
divictioCTDz <- merge(zdata, divictioCTD, by.x=c("to_idoperacion_num"),
by.y=c("id_operation")) # dim 650 obs x 56 vbles
```

---- ARRANGING COLUMNS AND ROWS ----

```
head(divictioCTDz)
```

```
divictioCTDz <- divictioCTDz[moveme(names(divictioCTDz), "estacion before lat")]
```

```
divictioCTDz <- divictioCTDz[moveme(names(divictioCTDz), "to_profundidad_operacion after estacion")]
```

divictioCTDz <- divictioCTDz %>% arrange(year)

---- FINAL FILE ----

```
write.xlsx(divictioCTDz, file="04.divictioCTDz.xlsx")
```

---- TUNA DIVERSITY DATA ----

```
divtuna  
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/03.tuna  
_div.xlsx")
```

```
divtuna <- data.frame(divtuna)
```

```
names(divtuna)
```

```
str(divtuna)
```

```
divtuna <- divtuna %>%
```

```
dplyr:: select(-...1, -horaGMT, -colector, -orden_bd) %>%
```

```
mutate_at(vars(orden, survey, estacion), funs(factor))
```

---- COMUN ELEMENTS DIVERSITY & CTD ----

```
comun <- factor(sort(intersect(divtuna$orden, myCTD$id_operation))) # 650 operaciones en  
comun
```

Las variables que use para fusionar tiene que tener datos en comun y ser factores

Los niveles de los factores comunes deben tener el mismo nombre

```
levels(divtuna$survey)
```

```
levels(myCTD$survey)
```

Fusiono tabla de diversidad con tabla de CTD segun numero de operacion y campana,

por estacion NO pq se repiten en las distintas campanas

```
divtunaCTD <- merge(myCTD, divtuna, by.x=c("id_operation", "survey"),
by.y=c("orden", "survey")) # dim 650 obs x 55 vbles
```

---- COMUN ELEMENTS divCTD & ZDATA ----

```
comun1 <- factor(sort(intersect(divtunaCTD$id_operation, zdata$to_idoperacion_num))) # 650
operaciones en comun
```

Fusiono tabla de diversidad con tabla de CTD segun numero de operacion y campana,

por estacion NO pq se repiten en las distintas campanas

```
divtunaCTDz <- merge(zdata, divtunaCTD, by.x=c("to_idoperacion_num"),
by.y=c("id_operation")) # dim 650 obs x 56 vbles
```

---- ARRANGING COLUMNS AND ROWS ----

```
head(divtunaCTDz)
```

```
divtunaCTDz <- divtunaCTDz[moveme(names(divtunaCTDz), "estacion before lat")]
```

```
divtunaCTDz <- divtunaCTDz[moveme(names(divtunaCTDz), "to_profundidad_operacion
after estacion")]
```

```
divtunaCTDz <- divtunaCTDz %>% arrange(year)
```

---- FINAL FILE ----

```
write.xlsx(divtunaCTDz, file="04.divtunaCTDz.xlsx")
```

---- MERGING CTD + COMPLETE + ICTIO + TUNA DIVERSITY ----

---- divCTDz (complete) + divictioCTDz (ictio) = df ----

```
comun <- factor(sort(intersect(divCTDz$to_idoperacion_num,
divictioCTDz$to_idoperacion_num))) # 650 operaciones en comun
```

```
df <- merge(divCTDz,
            divictioCTDz,
            by=c("to_idoperacion_num","survey","estacion","to_profundidad_operacion","lat","lon",
```

```
"year","month","day","day_of_year","time","timenorm","tow_duration",
```

```
"min_depth","max_depth","mld",
```

```
"max_fluor","depth_max_fluor","mean_fluor_10_100","cumul_fluor_10_100","mean_fluor_ml
d","cumul_fluor_mld",
```

```
"temp5","temp10","temp25","temp50","temp100","temp200","tmld",
```

```
"sal5","sal10","sal25","sal50","sal100","sal200","smlid",
```

```
"ox5","ox10","ox25","ox50","ox100","ox200","omld",
```

```
"gvel_u","gvel_v","gvel","meanChl","chl_month","chl_week","vsi","tmldR","Volumen"))
```

---- df + divtunaCTDz (tuna) = df1 ----

```
comun1 <- factor(sort(intersect(df$to_idoperacion_num, divtunaCTDz$to_idoperacion_num)))  
# 650 operaciones en comun
```

```
df1 <- merge(df,
            divtunaCTDz,
            by=c("to_idoperacion_num","survey","estacion","to_profundidad_operacion","lat","lon",
```

```
"year","month","day","day_of_year","time","timenorm","tow_duration",
```

```
"min_depth","max_depth","mld",
```

```
"max_fluor","depth_max_fluor","mean_fluor_10_100","cumul_fluor_10_100","mean_fluor_ml
d","cumul_fluor_mld",
```

```
"temp5","temp10","temp25","temp50","temp100","temp200","tmld",
```

```
"sal5","sal10","sal25","sal50","sal100","sal200","smlid",
```

```
"ox5","ox10","ox25","ox50","ox100","ox200","omld",
```

```
"gvel_u","gvel_v","gvel","meanChl","chl_month","chl_week","vsi","tmldR","Volumen"))
```

```
df1 <- df1 %>% arrange(year)
```

```
# ---- Rename dataset and variables ----
```

```
df <- df1 %>%
```

```
rename(operation = to_idoperacion_num, st_depth = to_profundidad_operacion,  
       Shannon=Shannon.x, SpsNum=SpsNum.x, X1.Simpson=X1.Simpson.x,  
       inv_Simps=inv_Simps.x,  
       Shan1=Shannon.y, SpsNum1=SpsNum.y, X1.Simp1=X1.Simpson.y,  
       inv_Simp1=inv_Simps.y,  
       Shan2=Shannon, SpsNum2=SpsNum, X1.Simp2=X1.Simpson, inv_Simp2=inv_Simps)
```

```
# ---- Final dataset for GAM analysis ----
```

```
write.xlsx(df, file="04.DIV_CTD.xlsx")
```

```
# ----
```

```
# ---- [[[[[[[[[[[ THE END ]]]]]]]]]]]]]]] ----
```

5. HOM_DATA.R

---- DIVERPEL: HOMOGENEOUS DATA ----

Author: ESTHER BARBER

Created on: 2020-01-30

Last modification: 2020-02-27

Remove westernmost sampling station to avoid Cap de la Nao

Remove night samples

Remove max_depth outliers

Remove duplicated stations

Final file with homogeneous data for GAM analysis --> 05.Hom_DATA.xlsx

Statistics summary of homogeneous data --> 05.Hom_DATA_summary.xlsx

---- PACKAGES ----

library(tidyverse)

library(xlsx)

library(dplyr)

library(readxl)

library(reshape2)

---- DATA ----

```
data
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/04.DIV
_CTD.xlsx")
```

```
data <- data.frame(data)
```

```
str(data)
```

```
data <- data %>%
```

```
  mutate_at(vars(operation, survey, estacion, year), funs(factor))
```

```
# dim(data) = 650 obs x 64 vbles
```

```
# ----
```

---- REMOVING WESTERN-MOST STATIONS ----

```
data %>%
```

```
  group_by(year) %>%
```

```
  summarize(minlon = min(lon))
```

```
df <- data[data$lon > 0.77 , ] # dim(df) = 639 obs x 64 vbles
```

```
df %>%
```

```
  group_by(year) %>%
```

```
  summarize(minlon = min(lon))
```

---- REMOVING NIGHT SAMPLES ----

```
df %>%
```

```
  group_by(year) %>%
```

```
  summarize(mintime = min(timenorm), maxtime=max(timenorm))
```

```
df <- df[df$timenorm > 0.243 & df$timenorm < 0.865 , ] # dim(df) = 571 obs x 64 vbles
```

---- REMOVING MAX_DEPTH OUT OF NORMAL SAMPLING DEPTHS ----

```
df %>%
```

```
group_by(year) %>%
  summarize(minz = min(max_depth, na.rm=T), maxz=max(max_depth, na.rm=T))
df <- df[df$max_depth > 0 & df$max_depth < 53 , ] # dim(df) = 567 obs x 64 vbles
```

---- REMOVING DUPLICATED STATIONS ----

which stations were sampled more than once?

```
x <- data %>%
```

```
group_by(year)%>%
```

```
count(estacion) %>% filter(n > 1)
```

```
x <- data.frame(x)
```

```
write.xlsx(x, file="04.St_duplo.xlsx")
```

---- FINAL HOMOGENEOUS DATASET ----

```
write.xlsx(df, file="05.Hom_DATA.xlsx")
```

Use the 04.St_duplo.xlsx file to identify duplicated sampled stations

And remove them from the file 05.Hom_DATA.xlsx by hand

Keep the first sampled stations from the ones duplicated

---- SUMMARY DATA ----

```
data <-  

read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/05.Hom  

_DATA.xlsx") # 548 obs x 64 vbles  

data <- data.frame(data)  

data <- data %>%  

  mutate_at(vars(operation, survey, estacion, year), funs(factor))
```



PROGRAMA
pleamar



Unión Europea
Fondo Europeo Marítimo y
de Pesca (FEMP)



6. DIV+CTD+DENS.R

```
#
```

```
# ---- DENSITY AND RELATIVE LARVAL DENSITY BY YEAR ----
```

```
# Author: Esther Barber
```

```
# Created on: December 2019
```

```
# Modified on: 9th Jan 2020 following
```

```
# https://stackoverflow.com/questions/37081612/calculate-relative-abundance-by-row-label-in-r-vegan-package
```

```
# Calculate larvae density * 100 m3 of each sps in each sample
```

```
# Merge DIV + CTD dataset with DENS dataset
```

```
# Exploring sps proportions per samples and years --> PhylaByYear2.xlsx
```

```
# Final file --> 06.DIV_CTD_DENS.xlsx
```

```
# _____
```

```
# ---- PACKAGES ----
```

```
library(tidyverse)
```

```
library(readxl)
```

```
library(ggplot2)
```

```
library(xlsx)
```

```
library(reshape2)
```

```
library(tidyverse)
```

```
library(dbplyr)
```

```
library(dplyr)
```

```
library(ggplot2)
```

```
# ----
```

---- ABUNDANCE DATA ----

```

abu <-  

read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/01.ictiot  

una.xlsx")  
  

abu <- data.frame(abu)  
  

names(abu)  
  

str(abu)  
  

abu <- abu %>%  
  

dplyr::select(-tc_id_colector_autonum, -tc_codestacion, -tc_orden_bd, -tp_HoraInicoGMT)  

%>%  
  

  mutate_at(vars(tc_campania, to_idoperacion_num), funs(factor))  
  

head(abu)

```

---- FROM ABUNDANCE DATA TO DENSITY DATA ----

```

# Calcula la densidad de cada taxón en 100m³  
  

dens_df <- with(abu, cbind(tc_campania, to_idoperacion_num, abu[!names(abu) %in%  

  c("tc_campania", "to_idoperacion_num")]/tc_m3*100))

```

---- DIV + CTD DATA ----

```

data <-  

read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/05.Hom  

_DATA.xlsx") # 548 obs x 64 vbles  
  

data <- data.frame(data)  
  

data <- data %>%  
  

  mutate_at(vars(operation, survey, estacion, year), funs(factor))

```

---- MERGING DENSITY DATA + CTD + DIVERSITY DATA ----

```
comun <- factor(sort(intersect(data$operation, dens_df$to_idoperacion_num))) # 548
operaciones en comun
```

Las variables que use para fusionar tiene que tener datos en comun y ser factores

Los niveles de los factores comunes deben tener el mismo nombre

```
levels(data$survey)
```

```
levels(dens_df$tc_campagna)
```

Fusiono tabla de diversidad+CTD con tabla de Densidad segun numero de operacion y campagna,

```
df1 <- merge(data, dens_df, by.x=c("survey", "operation"),
by.y=c("tc_campagna", "to_idoperacion_num")) # dim 548 obs x 200 vbles
```

---- FINAL FILE DIV+CTD+DENS ----

```
write.xlsx(df1, file="06.DIV_CTD_DENS.xlsx")
```

---- PLOTS ----

---- SPS RELATIVE DENSITY BY YEAR ----

Creo una nueva variable que es la suma de las densidades de larvas en cada muestra

```
df1$dens_ope <- rowSums(df1[, c(66:200)])
```

sum(df1\$dens_ope) # 32530.07 = densidad total de larvas de todo el dataset

para poder calcular la densidad relativa de larvas

---- Densidad relativa de cada especie por año y gráfico ----

```

rel_dens <- function(x){

  for(i in 2012:2016){

    temp <- x %>%
      filter(year == i)

    df <- (colSums(temp[, c(66:200)]))/(sum(temp$dens_ope))

    stack(df)

    df <- setNames(stack(df)[2:1], c('Species','RelDens'))

    p <- ggplot(df) +
      aes(x = Species, weight = RelDens) +
      geom_bar(fill = "#0c4c8a") +
      theme_bw() +
      theme(axis.title.x=element_blank(),
            axis.text.x=element_text(angle=90, vjust=0.5, size=9),
            plot.title = element_text(hjust=0.5)) +
      ggtitle(paste0("Relative density ",i)) +
      png(filename      =      paste0("C://Users/esther.barber/Desktop/DIVERPEL",i,"Relative
density_new1.png"), width = 30, height = 15, units = "cm", res = 400)
      print(p)
      dev.off()
    }

  rel_dens(df1)

# ---- SPS DENSITY BY YEAR ----

# density of each phyla by year

```

```

df2 <- df1[, c(7, 66:200)] %>%
  group_by(year) %>%
  summarise_all(sum) # sumar la densidad de cada taxon por año

# from wide shape to long shape
df3 <- melt(df2, id=c("year"), variable.name = "Phyla", value.name = "Density")

# Creating a factor variable to distinguish tuna sps
df3$group <- "tuna"
df3[1:min(650, nrow(df3)),]$group <- "ictio"
df3$group <- factor(df3$group, levels = c("tuna", "ictio"))
write.xlsx(df3, file="PhylaByYear.xlsx")

# ---- PLOTS ----

tiff("PhylaByYear.tiff", units="cm", width=18, height=14, res=300, compression="lzw")
ggplot(data= df3,
  aes(x=year, y=Density, fill=group)) +
  geom_bar(stat="identity", colour="black") +
  geom_text (aes(label=n, group=group), position = position_stack(vjust = .5),size=5) +
  ggtitle("Number Species by Year") +
  scale_fill_manual(values=c("#fdc926", "#0c4c8a")) +
  theme_bw() +
  theme(legend.title=element_blank(),
    legend.position="top",
    legend.text=element_text(size=16),
    plot.title = element_text(hjust=.5, size=30),
    axis.title=element_blank(),

```

```
axis.text.y = element_blank(),  
axis.ticks = element_blank(),  
axis.text.x = element_text(size=18))  
  
dev.off()
```

```
tiff("IctioByYear.tiff", units="cm", width=18, height=14, res=300, compression="lzw")  
  
ggplot(data= PhylaByYear[PhylaByYear$Density...0 == "TRUE" & PhylaByYear$group ==  
"ictio",],  
  
aes(x=Year, y=n)) +  
  
geom_bar(stat="identity", colour="black", fill= "#0c4c8a") +  
  
geom_text (aes(label=n), position = position_stack(vjust = .5),size=5, colour="white") +  
  
ggtitle("Number Species by Year") +  
  
theme_bw() +  
  
theme(plot.title = element_text(hjust=.5, size=30),  
  
axis.title=element_blank(),  
axis.text.y = element_blank(),  
axis.ticks = element_blank(),  
axis.text.x = element_text(size=18))  
  
dev.off()
```

7. SHANNONgam_200224.R:

```
# ---- DIVERPEL: GAM ANALYSIS FOR HOMOGENEOUS DATA ----
```

```
# Author: ESTHER BARBER
```

```
# Created on: 2020-01-30
```

```
# Last modification: 2020-02-24
```

```
# GAM plots with ggplot
```

```
# https://stackoverflow.com/questions/49471300/gam-plots-with-ggplot
```

```
# _____
```

```
# ---- PACKAGES ----
```

```
library(tidyverse)
```

```
library(xlsx)
```

```
library(dplyr)
```

```
library(readxl)
```

```
library(reshape2)
```

```
library(psych)
```

```
# Packages GAM
```

```
library(mgcv)
```

```
library(broom)
```

```
# Packages MAPS
```

```
library(MBA)
```

```
library(colorRamps)
```

```
library(maps)
```

```
library(maptools)
```

```
library(raster)
```

```
# ----
```

```
# ---- FUNCTIONS ----
```

```
# ---- transparent colors ----
```

```
# make transparent colors for overly points
```

```
makeTransparent<-function(someColor, alpha=100)
```

```
{
```

```
newColor<-col2rgb(someColor)
```

```
apply(newColor, 2, function(curcoldata){rgb(red=curcoldata[1], green=curcoldata[2],  

  blue=curcoldata[3],alpha=alpha, maxValue=255)})
```

```
}
```

```
# ---- upper.panel function ----
```

```
# Display correlation coefficient in pair scatterplots
```

```
upper.panel<-function(x, y){
```

```
  points(x,y, pch=19, col=makeTransparent(c("red", "green3", "blue", "purple",  

  "orange"))[data$year]) # coloured by factor Year
```

```
  r <- round(cor(x, y, use="complete.obs"), digits=2) # use="complete.obs" to avoid NA's
```

```
  txt <- paste0("R = ", r)
```

```
  usr <- par("usr"); on.exit(par(usr))
```

```
  par(usr = c(0, 1, 0, 1))
```

```
  text(0.5, 0.9, txt)}
```

```
# ---- myvis.gam function ----
```

myvis.gam function is a modification of vis.gam function to display nicer gradient colors in contour maps

```
#   https://stackoverflow.com/questions/21750020/changing-the-colors-in-a-contour-plot-from-vis-gam-in-mgcv
```

```
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF", "cyan",
" #7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))
```

```
myvis.gam <- function (x, view = NULL, cond = list(), n.grid = 30, too.far = 0,
```

```
col = NA, color = "heat", contour.col = NULL, se = -1, type = "link",
```

```
plot.type = "persp", zlim = NULL, nCol = 50, ...)
```

```
{
```

```
fac.seq <- function(fac, n.grid) {
```

```
fn <- length(levels(fac))
```

```
gn <- n.grid
```

```
if (fn > gn)
```

```
mf <- factor(levels(fac))[1:gn]
```

```
else {
```

```
ln <- floor(gn/fn)
```

```
mf <- rep(levels(fac)[fn], gn)
```

```
mf[1:(ln * fn)] <- rep(levels(fac), rep(ln, fn))
```

```
mf <- factor(mf, levels = levels(fac))
```

```
}
```

```
mf
```

```
}
```

```
dnm <- names(list(...))
```

```
v.names <- names(x$var.summary)
```

```
if (is.null(view)) {
```

```
k <- 0
```

```
view <- rep("", 2)
```

```
for (i in 1:length(v.names)) {
```

ok <- TRUE

```
if (is.matrix(x$var.summary[[i]]))
```

ok <- FALSE

```
else if (is.factor(x$var.summary[[i]])) {
```

```
  if (length(levels(x$var.summary[[i]])) <= 1)
```

ok <- FALSE

}

```
else {
```

```
  if (length(unique(x$var.summary[[i]])) == 1)
```

ok <- FALSE

}

```
  if (ok) {
```

k <- k + 1

```
    view[k] <- v.names[i]
```

}

```
  if (k == 2)
```

break

}

```
if (k < 2)
```

```
  stop("Model does not seem to have enough terms to do anything useful")
```

}

```
else {
```

```
  if (sum(view %in% v.names) != 2)
```

```
    stop(paste(c("view variables must be one of", v.names),
```

collapse = ", ")))

```
  for (i in 1:2) if (!inherits(x$var.summary[[view[i]]]),
```

c("numeric", "factor"))

```
    stop("Don't know what to do with parametric terms that are not simple numeric or factor  
variables")
```

}

ok <- TRUE

```
for (i in 1:2) if (is.factor(x$var.summary[[view[i]]])) {
```

```
    if (length(levels(x$var.summary[[view[i]]])) <= 1)
```

```
        ok <- FALSE
```

}

```
else {
```

```
    if (length(unique(x$var.summary[[view[i]]])) <= 1)
```

```
        ok <- FALSE
```

}

```
if (!ok)
```

```
stop(paste("View variables must contain more than one value. view = c(",
```

```
            view[1], ", ", view[2], ". ", sep = ""))
```

```
if (is.factor(x$var.summary[[view[1]]]))
```

```
m1 <- fac.seq(x$var.summary[[view[1]]], n.grid)
```

```
else {
```

```
r1 <- range(x$var.summary[[view[1]]])
```

```
m1 <- seq(r1[1], r1[2], length = n.grid)
```

}

```
if (is.factor(x$var.summary[[view[2]]]))
```

```
m2 <- fac.seq(x$var.summary[[view[2]]], n.grid)
```

```
else {
```

```
r2 <- range(x$var.summary[[view[2]]])
```

```
m2 <- seq(r2[1], r2[2], length = n.grid)
```

}

```
v1 <- rep(m1, n.grid)
```

```
v2 <- rep(m2, rep(n.grid, n.grid))
```

```
newd <- data.frame(matrix(0, n.grid * n.grid, 0))
```

```
for (i in 1:length(x$var.summary)) {
```

```

ma <- cond[[v.names[i]]]

if (is.null(ma)) {

  ma <- x$var.summary[[i]]

  if (is.numeric(ma))

    ma <- ma[2]

}

if (is.matrix(x$var.summary[[i]]))

  newd[[i]] <- matrix(ma, n.grid * n.grid, ncol(x$var.summary[[i]]),

    byrow = TRUE)

else newd[[i]] <- rep(ma, n.grid * n.grid)

}

names(newd) <- v.names

newd[[view[1]]] <- v1

newd[[view[2]]] <- v2

if (type == "link")

  zlab <- paste("linear predictor")

else if (type == "response")

  zlab <- type

else stop("type must be \"link\" or \"response\"")

fv <- predict.gam(x, newdata = newd, se.fit = TRUE, type = type)

z <- fv$fit

if (too.far > 0) {

  ex.tf <- exclude.too.far(v1, v2, x$model[, view[1]],

    x$model[, view[2]], dist = too.far)

  fv$se.fit[ex.tf] <- fv$fit[ex.tf] <- NA

}

if (is.factor(m1)) {

  m1 <- as.numeric(m1)

  m1 <- seq(min(m1) - 0.5, max(m1) + 0.5, length = n.grid)
}

```

}

```

if (is.factor(m2)) {

  m2 <- as.numeric(m2)

  m2 <- seq(min(m1) - 0.5, max(m2) + 0.5, length = n.grid)

}

if (se <= 0) {

  old.warn <- options(warn = -1)

  av <- matrix(c(0.5, 0.5, rep(0, n.grid - 1)), n.grid,
               n.grid - 1)

  options(old.warn)

  max.z <- max(z, na.rm = TRUE)

  z[is.na(z)] <- max.z * 10000

  z <- matrix(z, n.grid, n.grid)

  surf.col <- t(av) %*% z %*% av

  surf.col[surf.col > max.z * 2] <- NA

  if (!is.null(zlim)) {

    if (length(zlim) != 2 || zlim[1] >= zlim[2])

      stop("Something wrong with zlim")

    min.z <- zlim[1]

    max.z <- zlim[2]

  }

  else {

    min.z <- min(fv$fit, na.rm = TRUE)

    max.z <- max(fv$fit, na.rm = TRUE)

  }

  surf.col <- surf.col - min.z

  surf.col <- surf.col/(max.z - min.z)

  surf.col <- round(surf.col * nCol)

  con.col <- 1
}

```

```

if (color == "heat") {

  pal <- heat.colors(nCol)

  con.col <- 3

}

else if (color == "topo") {

  pal <- topo.colors(nCol)

  con.col <- 2

}

else if (color == "cm") {

  pal <- cm.colors(nCol)

  con.col <- 1

}

else if (color == "terrain") {

  pal <- terrain.colors(nCol)

  con.col <- 2

}

else if (color == "gray" || color == "bw") {

  pal <- gray(seq(0.1, 0.9, length = nCol))

  con.col <- 1

}

#### customized here

else if (color == 'jet') {

  pal <- jet.colors(nCol)

  con.col = 1

}

#####

else stop("color scheme not recognised")

if (is.null(contour.col))

  contour.col <- con.col

```

```

surf.col[surf.col < 1] <- 1

surf.col[surf.col > nCol] <- nCol

if (is.na(col))

col <- pal[as.array(surf.col)]

z <- matrix(fv$fit, n.grid, n.grid)

if (plot.type == "contour") {

  stub <- paste(ifelse("xlab" %in% dnm, "", ",xlab=view[1]"),

  ifelse("ylab" %in% dnm, "", ",ylab=view[2]"),

  ifelse("main" %in% dnm, "", ",main=zlab"), "...)",

  sep = "")

  if (color != "bw") {

    txt <- paste("image(m1,m2,z,col=col,zlim=c(min.z,max.z)",

    stub, sep = "")

    eval(parse(text = txt))

    txt <- paste("contour(m1,m2,z,col=contour.col,zlim=c(min.z,max.z)",

    ifelse("add" %in% dnm, "", ",add=TRUE"), "...)",

    sep = "")

    eval(parse(text = txt))

  }

  else {

    txt <- paste("contour(m1,m2,z,col=1,zlim=c(min.z,max.z)",

    stub, sep = "")

    eval(parse(text = txt))

  }

}

else {

  stub <- paste(ifelse("xlab" %in% dnm, "", ",xlab=view[1]"),

  ifelse("ylab" %in% dnm, "", ",ylab=view[2]"),

  ifelse("main" %in% dnm, "", ",zlab=zlab"), "...)",
```

```

sep = "")}

if (color == "bw") {

  op <- par(bg = "white")

  txt <- paste("persp(m1,m2,z,col=|"white|",zlim=c(min.z,max.z) ",

    stub, sep = "")

  eval(parse(text = txt))

  par(op)

}

else {

  txt <- paste("persp(m1,m2,z,col=col,zlim=c(min.z,max.z)",

    stub, sep = "")

  eval(parse(text = txt))

}

}

else {

  if (color == "bw" || color == "gray") {

    subs <- paste("grey are +/-", se, "s.e.")

    lo.col <- "gray"

    hi.col <- "gray"

  }

  else {

    subs <- paste("red/green are +/-", se, "s.e.")

    lo.col <- "green"

    hi.col <- "red"

  }

  if (!is.null(zlim)) {

    if (length(zlim) != 2 || zlim[1] >= zlim[2])

      stop("Something wrong with zlim")
  }
}

```

```

min.z <- zlim[1]

max.z <- zlim[2]

}

else {

z.max <- max(fv$fit + fv$se.fit * se, na.rm = TRUE)

z.min <- min(fv$fit - fv$se.fit * se, na.rm = TRUE)

}

zlim <- c(z.min, z.max)

z <- fv$fit - fv$se.fit * se

z <- matrix(z, n.grid, n.grid)

if (plot.type == "contour")

warning("sorry no option for contouring with errors: try plot.gam")

stub <- paste(ifelse("xlab" %in% dnm, "", ",xlab=view[1]"),

ifelse("ylab" %in% dnm, "", ",ylab=view[2]"), ifelse("zlab" %in% 

dnm, "", ",zlab=zlab"), ifelse("sub" %in% dnm, 

"", ",sub=subs"), "...)", sep = 

"))

txt <- paste("persp(m1,m2,z,col=col,zlim=zlim", ifelse("border" %in% 

dnm, "", ",border=lo.col"), stub, sep = "")

eval(parse(text = txt))

par(new = TRUE)

z <- fv$fit

z <- matrix(z, n.grid, n.grid)

txt <- paste("persp(m1,m2,z,col=col,zlim=zlim", ifelse("border" %in% 

dnm, "", ",border=\"black\""), stub, sep = "")

eval(parse(text = txt))

par(new = TRUE)

z <- fv$fit + se * fv$se.fit

z <- matrix(z, n.grid, n.grid)

```

```
txt <- paste("persp(m1,m2,z,col=col,zlim=zlim", ifelse("border" %in%
dnm, "", ",border=hi.col"), stub, sep = "")
```

```
eval(parse(text = txt))
```

```
}
```

```
}
```

```
# Calling function defined in R
```

```
#source("C:/Users/esther.barber/Desktop/R software/LM and GLM summary export.R")
```

```
# _____
```

```
# ----
```

```
# _____
```

```
# ---- GAM DATA ----
```

```
data <-  
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/08.abu  
GAM.xlsx")
```

```
data <- data.frame(data)
```

```
str(data)
```

```
data <- data %>%
```

```
mutate_at(vars(operation, survey, estacion, year), funs(factor))
```

```
# dim(data) = 650 obs x 64 vbles
```

```
# ---- SCATTERPLOT MATRIX PLOT GEO DATA w/SHANNON INDEX FOR  
COMPLETE, ICTIO AND TUNA DATASETS ----
```

```
# st_depth[4]; lat[5]; lon[6]; timenorm[12]; max_depth[15]; Shannon[53]; Shan1[57];  
Shan2[61]
```

```
# Version 1
```

Correlation upper panel customized

Create the plots

```
pairs(data[, c(4:6, 12, 15, 53, 57, 61)],
```

```
lower.panel = NULL,
```

```
upper.panel = upper.panel)
```

Version 2

```
pairs.panels(data[, c(4:6, 12, 15, 53, 57, 61)],
```

```
method = "pearson", # correlation method
```

```
hist.col = "#00AFBB",
```

```
density = TRUE, # show density plots
```

```
ellipses = TRUE) # show correlation ellipses
```

---- SHANNON HISTOGRAMS BY YEAR ----

Complete data

```
ggplot(data) +
```

```
aes(x = Shannon) +
```

```
geom_histogram(bins=45L, fill = "#0c4c8a", color="black") +
```

```
labs(title = "Shannon COMPLETE") +
```

```
scale_x_continuous(limits = c(-0.05, 2.9),
```

```
breaks=seq(from=0, to=2.8, by=0.25), expand=c(0,0)) +
```

```
theme_bw() +
```

```
theme(plot.title = element_text(hjust=0.5)) +
```

```
facet_grid(rows=vars(year))
```

Ictioplankton data

```
ggplot(data) +
```

```

aes(x = Shan1) +
  geom_histogram(bins=45L, fill = "#0c4c8a", color="black") +
  labs(title = "Shannon ICTIO") +
  scale_x_continuous(limits = c(-0.05, 2.9),
                     breaks=seq(from=0, to=2.9, by=0.25), expand=c(0,0)) +
  theme_bw() +
  theme(plot.title = element_text(hjust=0.5)) +
  facet_grid(rows=vars(year))

```

Tuna data

```

ggplot(data) +
  aes(x = Shan2) +
  geom_histogram(bins=30L, fill = "#0c4c8a", color="black") +
  labs(title = "Shannon TUNA") +
  scale_x_continuous(limits = c(-0.05, 1.5),
                     breaks=seq(from=0, to=1.5, by=0.1), expand=c(0,0)) +
  theme_bw() +
  theme(plot.title = element_text(hjust=0.5)) +
  facet_grid(rows=vars(year))

```

---- GENERAL MAP ----

Load maps data:

```

spain <- getData("GADM", country = "ESP", level = 0)
# Map limits
xlim <- c(min(data$lon, na.rm=F), max(data$lon, na.rm=F))

```

```
ylim <- c(min(data$lat, na.rm=F), max(data$lat, na.rm=F))
```

```
# ----
```

```
# _____
```

```
# -----
```

```
# ---- BASIC GAM ----
```

```
# -----
```

```
# ---- SHANNON ~ POS ----
```

```
# ---- GAM1 = Complete data ----
```

```
gam1 <- gam(Shannon ~ s(lat, lon, k=15) + factor(year) , data=data)
```

```
summary(gam1)
```

```
# PLOT GAM1
```

```
# Map from the model
```

```
myvis.gam(gam1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

```
# overlay map on model's map
```

```
par(new=T)
```

```
# balearic island map with transparent background
```

```
plot(spanish, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D", sub="COMPLETE",  
font.sub=2, col.sub="red") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

```
# Overly sampling stations positions
```

```
points(data$lon, data$lat, pch=19, col=makeTransparent("black"), cex=.3)
```

```
# _____
```

---- GAM1.1 = Ictioplankton data ----

```
gam1.1 <- gam(Shan1 ~ s(lat, lon, k=15) + factor(year) , data=data)
summary(gam1.1)
```

PLOT GAM1.1

Map from the model

```
myvis.gam(gam1.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

overlay map on model's map

```
par(new=T)
```

balearic island map with transparent background

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D", sub="ICTIO", font.sub=2,
col.sub="red") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

Overly sampling stations positions

```
points(data$lon, data$lat, pch=19, col=makeTransparent("black"), cex=.3)
```

```
# _____
```

---- SHANNON ~ ST_Z ----

---- GAM 2 = Complete data ----

```
gam2 <- gam(Shannon ~ s(log(st_depth), k=3) + factor(year) , data=data)
summary(gam2)
```

PLOT GAM 2

```
par(mfrow=c(1,2))
```

partial effect of station depth

```
plot(gam2, se=T, select=1, shade=T, scale=0, rug=T, res=T, pch=1, xlab="log st_depth",
sub="COMPLETE", font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
plot(gam2, se=T, select=1, shade=T, scale=0, rug=T, xlab="log st_depth", sub="COMPLETE",
font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
# -----
```

```
# ---- GAM 2.1 = Ictioplankton data ----
```

```
gam2.1 <- gam(Shan1 ~ s(log(st_depth), k=3) + factor(year), data=data)
```

```
summary(gam2.1)
```

```
# PLOT GAM 2.1
```

```
par(mfrow=c(1,2))
```

```
# partial effect of station depth
```

```
plot(gam2.1, se=T, select=1, shade=T, scale=0, rug=T, res=T, pch=1, xlab="log st_depth",
sub="ICTIO", font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
plot(gam2.1, se=T, select=1, shade=T, scale=0, rug=T, xlab="log st_depth", sub="ICTIO",
font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
# -----
```

```
# ---- SHANNON ~ TIME ----
```

```
# ---- GAM 3 = Complete data ----
```

```
gam3 <- gam(Shannon ~ s(timenorm, bs="cc", k=4) + factor(year), data=data)
```

```
summary(gam3)
```

```
# PLOT GAM 3
```

```
par(mfrow=c(1,2))
```

partial effect of station depth

```
plot(gam3, se=T, select=1, shade=T, scale=0, rug=T, res=T, pch=1, xlab="timenorm",
sub="COMPLETE", font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
plot(gam3, se=T, select=1, shade=T, scale=0, rug=T, xlab="timenorm", sub="COMPLETE",
font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
# _____
```

---- GAM 3.1 = Ictioplankton data ----

```
gam3.1 <- gam(Shan1 ~ s(timenorm, bs="cc", k=4) + factor(year), data=data)
```

```
summary(gam3.1)
```

PLOT GAM 3.1

```
par(mfrow=c(1,2))
```

partial effect of station depth

```
plot(gam3.1, se=T, select=1, shade=T, scale=0, rug=T, res=T, pch=1, xlab="timenorm",
sub="ICTIO", font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
plot(gam3.1, se=T, select=1, shade=T, scale=0, rug=T, xlab="timenorm", sub="ICTIO",
font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
# ----
```

---- SHANNON ~ MAX_Z ----

---- GAM 4 = Complete data ----

```
gam4 <- gam(Shannon ~ s(log(max_depth), k=3) + factor(year), data=data)
```

```
summary(gam4)
```

PLOT GAM 4

```

par(mfrow=c(1,2))

# partial effect of station depth

plot(gam4, se=T, select=1, shade=T,scale=0,rug=T,res=T, pch=1, xlab="log max_depth",
sub="COMPLETE", font.sub=2, col.sub="red")

abline(h=0,lty=2,col=2,lwd=2)

plot(gam4, se=T, select=1, shade=T,scale=0,rug=T, xlab="log max_depth",
sub="COMPLETE", font.sub=2, col.sub="red")

abline(h=0,lty=2,col=2,lwd=2)

# _____


# ---- GAM 4.1 = Ictioplankton data ----

gam4.1 <- gam(Shan1 ~ s(log(max_depth), k=3) , data=data)

summary(gam4.1)


# PLOT GAM 4.1

par(mfrow=c(1,2))

# partial effect of station depth

plot(gam4.1, se=T, select=1, shade=T,scale=0,rug=T,res=T, pch=1, xlab="log max_depth",
sub="ICTIO", font.sub=2, col.sub="red")

abline(h=0,lty=2,col=2,lwd=2)

plot(gam4.1, se=T, select=1, shade=T,scale=0,rug=T, xlab="log max_depth", sub="ICTIO",
font.sub=2, col.sub="red")

abline(h=0,lty=2,col=2,lwd=2)

# ----


# ---- SHANNON ~ POS + ST_Z ----

# ---- GAM2x = Complete data ----

gam2x <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) +
factor(year), data=data)

summary(gam2x)

```

```

# PLOT GAM2x

par(mfrow=c(1,2))

# Map from the model

myvis.gam(gam2x,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on model's map

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D", sub="COMPLETE",
font.sub=2, col.sub="red") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling stations positions

points(data$lon, data$lat, pch=19, col=makeTransparent("black"), cex=.3)

# Partial effecto of ST_Z

plot(gam2x, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", sub="COMPLETE",
font.sub=2, col.sub="red")

abline(h=0,lty=2,col=2,lwd=2)

# _____
```

---- GAM2x.1 = Ictioplankton data ----

```

gam2x.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3),
data=data)

summary(gam2x.1)

# PLOT GAM2x

par(mfrow=c(1,2))

# Map from the model

myvis.gam(gam2x.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

```

overlay map on model's map

```
par(new=T)
```

balearic island map with transparent background

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D", sub="ICTIO", font.sub=2,
col.sub="red") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

Overly sampling stations positions

```
points(data$lon, data$lat, pch=19, col=makeTransparent("black"), cex=.3)
```

Partial efecto of ST_Z

```
plot(gam2x.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", sub="ICTIO",
font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

---- SHANNON ~ POS + TIME ----

---- GAM3x = Complete data ----

```
gam3x <- gam(Shannon ~ s(lat, lon, k=15) + s(timenorm, bs="cc", k=4),
```

```
data=data)
```

```
summary(gam3x)
```

PLOT GAM3x

```
par(mfrow=c(1,2))
```

Map from the model

```
myvis.gam(gam3x,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

overlay map on model's map

```
par(new=T)
```

balearic island map with transparent background

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D", sub="COMPLETE",
font.sub=2, col.sub="red") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

```
# Overly sampling stations positions
```

```
points(data$lon, data$lat, pch=19, col=makeTransparent("black"), cex=.3)
```

```
# Partial efecto of TIME
```

```
plot(gam3x, se=T, select=2, shade=T, scale=0, rug=T, xlab="Time", sub="COMPLETE",  
font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
# _____
```

```
# ---- GAM3x.1 = Ictioplankton data ----
```

```
gam3x.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(timenorm, bs="cc", k=4),
```

```
data=data)
```

```
summary(gam3x.1)
```

```
# PLOT GAM3x.1
```

```
par(mfrow=c(1,2))
```

```
# Map from the model
```

```
myvis.gam(gam3x.1, view=c('lon','lat'), plot.type="contour", color="jet", too.far=0.05)
```

```
# overlay map on model's map
```

```
par(new=T)
```

```
# balearic island map with transparent background
```

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D", sub="ICTIO", font.sub=2,  
col.sub="red") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

```
# Overly sampling stations positions
```

```
points(data$lon, data$lat, pch=19, col=makeTransparent("black"), cex=.3)
```

```
# Partial efecto of TIME
```

```
plot(gam3x.1, se=T, select=2, shade=T, scale=0, rug=T, xlab="Time", sub="ICTIO", font.sub=2,
col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
# ----
```

```
# ---- SHANNON ~ POS + MAX_Z ----
```

```
# ---- GAM4x = Complete data ----
```

```
gam4x <- gam(Shannon ~ s(lat, lon, k=15) + s(log(max_depth), k=3), data=data)
```

```
summary(gam4x)
```

```
# PLOT GAM4x
```

```
par(mfrow=c(1,2))
```

```
# Map from the model
```

```
myvis.gam(gam4x, view=c('lon','lat'), plot.type="contour", color="jet", too.far=0.05)
```

```
# overlay map on model's map
```

```
par(new=T)
```

```
# balearic island map with transparent background
```

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D", sub="COMPLETE",
font.sub=2, col.sub="red") # rgb=c(0,0,0, alpha=0.05)
```

```
box()
```

```
# Overly sampling stations positions
```

```
points(data$lon, data$lat, pch=19, col=makeTransparent("black"), cex=.3)
```

```
# Partial efecto of MAX_Z
```

```
plot(gam4x, se=T, select=2, shade=T, scale=0, rug=T, xlab="Max_z", sub="COMPLETE",
font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

```
# _____
```

```
# ---- GAM4x.1 = Ictioplankton data ----
```

```
gam4x.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(max_depth), k=3), data=data)
```

```
summary(gam4x.1)
```

```
# PLOT GAM4x.1
```

```
par(mfrow=c(1,2))
```

```
# Map from the model
```

```
myvis.gam(gam4x.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

```
# overlay map on model's map
```

```
par(new=T)
```

```
# balearic island map with transparent background
```

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D", sub="ICIO", font.sub=2,
col.sub="red") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

```
# Overly sampling stations postitions
```

```
points(data$lon, data$lat, pch=19, col=makeTransparent("black"), cex=.3)
```

```
# Partial efecto of MAX_Z
```

```
plot(gam4x.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="Max_z", sub="ICTIO",
font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# ----
```

```
# ---- SHANNON ~ ST_Z + TIME ----
```

```
# ---- GAM5x = Complete data ----
```

```
gam5x <- gam(Shannon ~ s(log(st_depth), k=3) + s(timenorm, bs="cc", k=4),
```

```
data=data)
```

```
summary(gam5x)
```

PLOT GAM5x

```
par(mfrow=c(1,2))
```

Partial efect of ST_Z

```
plot(gam5x, se=T, select=1, shade=T, scale=0, rug=T, xlab="St_z", sub="COMPLETE",  
font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

Partial efecto of TIME

```
plot(gam5x, se=T, select=2, shade=T, scale=0, rug=T, xlab="Time", sub="COMPLETE",  
font.sub=2, col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

---- GAM5x.1 = Ictioplankton data ----

```
gam5x.1 <- gam(Shan1 ~ s(log(st_depth), k=3) + s(timenorm, bs="cc", k=4) +  
factor(year), data=data)  
summary(gam5x.1)
```

PLOT GAM5x

```
par(mfrow=c(1,2))
```

Partial efect of ST_Z

```
plot(gam5x.1, se=T, select=1, shade=T, scale=0, rug=T, xlab="St_z", sub="ICTIO", font.sub=2,  
col.sub="red")
```

```
abline(h=0, lty=2, col=2, lwd=2)
```

Partial efecto of TIME

```
plot(gam5x.1, se=T, select=2, shade=T, scale=0, rug=T, xlab="Time", sub="ICTIO", font.sub=2,  
col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# ----
```

```
# ---- SHANNON ~ ST_Z + MAX_Z ----
```

```
# ---- GAM6x = Complete data ----
```

```
gam6x <- gam(Shannon ~ s(log(st_depth), k=3) + s(log(max_depth), k=3) , data=data)
```

```
summary(gam6x)
```

```
# PLOT GAM6x
```

```
par(mfrow=c(1,2))
```

```
# Partial efect of ST_Z
```

```
plot(gam6x, se=T, select=1, shade=T,scale=0,rug=T, xlab="St_z", sub="COMPLETE",  
font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# Partial efecto of MAX_Z
```

```
plot(gam6x, se=T, select=2, shade=T,scale=0,rug=T, xlab="Max_z", sub="COMPLETE",  
font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# _____
```

```
# ---- GAM6x.1 = Ictioplankton data ----
```

```
gam6x.1 <- gam(Shan1 ~ s(log(st_depth), k=3) + s(log(max_depth), k=3) , data=data)
```

```
summary(gam6x.1)
```

```
# PLOT GAM6x.1
```

```
par(mfrow=c(1,2))
```

Partial efect of ST_Z

```
plot(gam6x.1, se=T, select=1, shade=T,scale=0,rug=T, xlab="St_z", sub="ICTIO", font.sub=2,
col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

Partial efecto of MAX_Z

```
plot(gam6x.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="Max_z", sub="ICTIO",
font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

---- SHANNON ~ TIME + MAX_Z ----

---- GAM7x = Complete data ----

```
gam7x <- gam(Shannon ~ s(timenorm, bs="cc", k=4) + s(log(max_depth), k=3) , data=data)
```

```
summary(gam7x)
```

PLOT GAM7x

```
par(mfrow=c(1,2))
```

Partial efect of TIME

```
plot(gam7x, se=T, select=1, shade=T,scale=0,rug=T, xlab="Time", sub="COMPLETE",
font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

Partial efecto of MAX_Z

```
plot(gam7x, se=T, select=2, shade=T,scale=0,rug=T, xlab="Max_z", sub="COMPLETE",
font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

---- GAM7x.1 = Ictioplankton data ----

```
gam7x.1 <- gam(Shan1 ~ s(timenorm, bs="cc", k=4) + s(log(max_depth), k=3) , data=data)
summary(gam7x.1)
```

PLOT GAM7x.1

```
par(mfrow=c(1,2))
```

Partial efect of TIME

```
plot(gam7x.1, se=T, select=1, shade=T,scale=0,rug=T, xlab="Time", sub="ICTIO", font.sub=2,
col.sub="red")
abline(h=0,lty=2,col=2,lwd=2)
```

Partial efecto of MAX_Z

```
plot(gam7x.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="Max_z", sub="ICTIO",
font.sub=2, col.sub="red")
abline(h=0,lty=2,col=2,lwd=2)
# ----
```

---- SHANNON ~ POS + ST_Z + TIME + MAX_Z ----

---- GAM 5 = Complete data ----

```
gam5 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(log(max_depth), k=3), data=data)
summary(gam5)
```

PLOT GAM 5

To display the plots on the same sheet

```
par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

Map from the model

```
myvis.gam(gam5,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

overlay map on map's model

```
par(new=T)
```

balearic island map with transparent background

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

Overly sampling LAT-LON

```
points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)
```

partial effect of ST_DEPTH

```
plot(gam5, se=T, select=2, shade=T, scale=0, rug=T, xlab="log st_depth", main="COMPLETE",
font.main=2, col.main="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

partial effect of TIMENORM

```
plot(gam5, se=T, select=3, shade=T, scale=0, rug=T, xlab="timenorm")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

partial effect of MAX_DEPTH

```
plot(gam5, se=T, select=4, shade=T, scale=0, rug=T, xlab="log max_depth")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

---- GAM 5.1 = Ictioplankton data ----

```
gam5.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(log(max_depth), k=3), data=data)
```

```
summary(gam5.1)
```

PLOT GAM 5.1

To display the plots on the same sheet

```
par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

```
# Map from the model
```

```
myvis.gam(gam5.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

```
# overlay map on map's model
```

```
par(new=T)
```

```
# balearic island map with transparent background
```

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

```
# Overly sampling LAT-LON
```

```
points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)
```

```
# partial effect of ST_DEPTH
```

```
plot(gam5.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO", font.main=2, col.main="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of TIMENORM
```

```
plot(gam5.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of MAX_DEPTH
```

```
plot(gam5.1, se=T, select=4, shade=T,scale=0,rug=T, xlab="log max_depth")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# ----
```

```
# ---- SHANNON ~ POS + ST_Z + TIME ----
```

```
# ---- GAM 6 = Complete data ----
```

```
gam6 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4),
```

```
data=data)
```

summary(gam6)

PLOT GAM 6

To display the plots on the same sheet

```
par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

Map from the model

```
myvis.gam(gam6,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

overlay map on map's model

```
par(new=T)
```

balearic island map with transparent background

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

Overly sampling LAT-LON

```
points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)
```

partial effect of ST_DEPTH

```
plot(gam6, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="COMPLETE",  

font.main=2, col.main="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

partial effect of TIMENORM

```
plot(gam6, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

---- GAM 6.1 = Ictioplankton data ----

```
gam6.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4),
```

```
data=data)
```

summary(gam6.1)

```
# PLOT GAM 6.1

# To display the plots on the same sheet

par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))

# Map from the model

myvis.gam(gam6.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on map's model

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling LAT-LON

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

plot(gam6.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO",
font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM

plot(gam6.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# ----

# ---- SHANNON ~ ST_Z + TIME + MAX_Z ----

# ---- GAM 66 = Complete data ----
```

```
gam66 <- gam(Shannon ~ s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(log(max_depth)), data=data)
```

```
summary(gam66)
```

```
# PLOT GAM 66
```

```
par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

```
# partial effect of ST_DEPTH
```

```
plot(gam66, se=T, select=1, shade=T,scale=0,rug=T, xlab="log st_depth",
main="COMPLETE", font.main=2, col.main="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of TIMENORM
```

```
plot(gam66, se=T, select=2, shade=T,scale=0,rug=T, xlab="timenorm")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of TIMENORM
```

```
plot(gam66, se=T, select=3, shade=T,scale=0,rug=T, xlab="log max_depth")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# ---- GAM 66.1 = Ictioplankton data ----
```

```
gam66.1 <- gam(Shan1 ~ s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(log(max_depth)), data=data)
```

```
summary(gam66.1)
```

```
# PLOT GAM 66.1
```

```
par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

```
# partial effect of ST_DEPTH
```

```
plot(gam66.1, se=T, select=1, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO",
font.main=2, col.main="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of TIMENORM
```

```
plot(gam66.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="timenorm")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of TIMENORM
```

```
plot(gam66.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="log max_depth")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# ----
```

```
# ---- SHANNON ~ POS + ST_Z + TIME + VSI ----
```

```
# ---- GAM 7 = Complete data ----
```

```
gam7 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(vsi, k=3) +
```

```
factor(year), data=data)
```

```
summary(gam7)
```

```
# PLOT GAM 7
```

```
# To display the plots on the same sheet
```

```
par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

```
# Map from the model
```

```
myvis.gam(gam7,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

```
# overlay map on map's model
```

```
par(new=T)
```

```
# balearic island map with transparent background
```

```

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling LAT-LON

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

plot(gam7, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="COMPLETE",
font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM

plot(gam7, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of VSI

plot(gam7, se=T, select=4, shade=T,scale=0,rug=T, xlab="strat.index")

abline(h=0,lty=2,col=2,lwd=2)

# ---- GAM 7.1 = Ictioplankton data ----

gam7.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(vsi, k=3) +
factor(year), data=data)

summary(gam7.1)

# PLOT GAM 7.1

# To display the plots on the same sheet

par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))

# Map from the model

```

```
myvis.gam(gam7.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on map's model

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling LAT-LON

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

plot(gam7.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO",
font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM

plot(gam7.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of VSI

plot(gam7.1, se=T, select=4, shade=T,scale=0,rug=T, xlab="strat. index")

abline(h=0,lty=2,col=2,lwd=2)

# ----

# -----
```



```
# ---- HYDRO variables ----
```



---- SCATTER PLOT SHANNON WITH HYDRO VARIABLES ----

```
# cumul_fluor_10_100[22]; tmld[29]; smld[36];
```

Version 1

```
# Correlation upper panel customized
```

```
# Create the plots
```

```
pairs(data[, c(22, 29, 36, 53, 57, 61)],
```

lower.panel = NULL,

upper.panel = upper.panel)

Version 2

```
pairs.panels(data[, c(22, 29, 36, 53, 57, 61)],
```

```
method = "pearson", # correlation method
```

```
hist.col = "#00AFBB",
```

```
density = TRUE, # show density plots
```

```
ellipses = TRUE) # show correlation ellipses
```

#

#

THE END

---- SHANNON ~ FLUO ----

---- GAM 8 = Complete data ----

```
gam8 <- gam(Shannon ~ s(cumul_fluor_10_100, k=3) + factor(year), data=data)
summary(gam8)
```

PLOT GAM 8

partial effect of station depth

```
par(mfrow=c(1,2))

plot(gam8, se=T, select=1, shade=T, scale=0, rug=T, res=T, pch=1, xlab="cumul_fluo_10_100",
sub="COMPLETE", font.sub=2, col.sub="red")

abline(h=0, lty=2, col=2, lwd=2)

plot(gam8, se=T, select=1, shade=T, scale=0, rug=T, xlab="cumul_fluo_10_100",
sub="COMPLETE", font.sub=2, col.sub="red")

abline(h=0, lty=2, col=2, lwd=2)

# _____
```

---- GAM 8.1 = Ictioplankton data ----

```
gam8.1 <- gam(Shan1 ~ s(cumul_fluor_10_100, k=3) +
factor(year), data=data)

summary(gam8.1)
```

PLOT GAM 8.1

partial effect of station depth

```
par(mfrow=c(1,2))

plot(gam8.1, se=T, select=1, shade=T, scale=0, rug=T, res=T, pch=1,
xlab="cumul_fluo_10_100", sub="ICTIO", font.sub=2, col.sub="red")

abline(h=0, lty=2, col=2, lwd=2)
```



```
plot(gam8.1, se=T, select=1, shade=T,scale=0,rug=T,  
sub="ICTIO", font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# _____
```

```
# ----
```

```
# ---- SHANNON ~ TMLD ----
```

```
# ---- GAM 9 = Complete data ----
```

```
gam9 <- gam(Shannon ~ s(tmld, k=3) + factor(year), data=data)
```

```
summary(gam9)
```

```
# PLOT GAM 9
```

```
# partial effect of station depth
```

```
par(mfrow=c(1,2))
```

```
plot(gam9, se=T, select=1, shade=T,scale=0,rug=T,res=T, pch=1, xlab="Temp.MLD",  
sub="COMPLETE", font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
plot(gam9, se=T, select=1, shade=T,scale=0,rug=T, xlab="Temp.MLD", sub="COMPLETE",  
font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# _____
```

```
# ---- GAM 9.1 = Ictioplankton data ----
```

```
gam9.1 <- gam(Shan1 ~ s(tmld, k=3) + factor(year) , data=data)
```

```
summary(gam9.1)
```

```
# PLOT GAM 9.1
```

partial effect of station depth

```
par(mfrow=c(1,2))

plot(gam9.1, se=T, select=1, shade=T,scale=0,rug=T,res=T, pch=1, xlab="Temp.MLD",
sub="ICTIO", font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
plot(gam9.1, se=T, select=1, shade=T,scale=0,rug=T, xlab="Temp.MLD", sub="ICTIO",
font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# _____
```

```
# ----
```

```
# ---- SHANNON ~ SMLD ----
```

```
# ---- GAM 10 = Complete data ----
```

```
gam10 <- gam(Shannon ~ s(smld, k=3) + factor(year) , data=data)
```

```
summary(gam10)
```

```
# PLOT GAM 10
```

partial effect of station depth

```
par(mfrow=c(1,2))
```

```
plot(gam10, se=T, select=1, shade=T,scale=0,rug=T,res=T, pch=1, xlab="Sal.MLD",
sub="COMPLETE", font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
plot(gam10, se=T, select=1, shade=T,scale=0,rug=T, xlab="Sal.MLD", sub="COMPLETE",
font.sub=2, col.sub="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# _____
```

```
# ---- GAM 10.1 = Ictioplankton data ----
```

```

gam10.1 <- gam(Shan1 ~ s(smld, k=3) + factor(year), data=data)

summary(gam10.1)

# PLOT GAM 10.1


# partial effect of station depth

par(mfrow=c(1,2))

plot(gam10.1, se=T, select=1, shade=T, scale=0, rug=T, res=T, pch=1, xlab="Sal.MLD",
sub="ICTIO", font.sub=2, col.sub="red")

abline(h=0,lty=2,col=2,lwd=2)

plot(gam10.1, se=T, select=1, shade=T, scale=0, rug=T, xlab="Sal.MLD", sub="ICTIO",
font.sub=2, col.sub="red")

abline(h=0,lty=2,col=2,lwd=2)

# ----

# ---- SHANNON ~ POS + ST_Z + TIME + FLUO ----

# ---- GAM 11 = Complete data ----

gam11 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(cumul_fluor_10_100, k=3), data=data)

summary(gam11)

# PLOT GAM 11


# To display the plots on the same sheet

par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))

# Map from the model

myvis.gam(gam11,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on map's model

```

```

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling LAT-LON

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

plot(gam11, se=T, select=2, shade=T, scale=0, rug=T, xlab="log st_depth",
main="COMPLETE", font.main=2, col.main="red")

abline(h=0, lty=2, col=2, lwd=2)

# partial effect of TIMENORM

plot(gam11, se=T, select=3, shade=T, scale=0, rug=T, xlab="timenorm")

abline(h=0, lty=2, col=2, lwd=2)

# partial effect of FLUORESCENCE

plot(gam11, se=T, select=4, shade=T, scale=0, rug=T, xlab="cumul_fluo_10_100")

abline(h=0, lty=2, col=2, lwd=2)

# ---- GAM 11.1 = Ictioplankton data ----

gam11.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc', k=4) +
  s(cumul_fluor_10_100, k=3),
  data=data)

summary(gam11.1)

# PLOT GAM 11.1

# To display the plots on the same sheet

par(mfrow=c(2,2), omi=c(0.2, 0.2, 0.2, 0.2), mai=c(0.7, 0.7, 0.2, 0.2))

# Map from the model

```

```

myvis.gam(gam11.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on map's model

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling LAT-LON

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

plot(gam11.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO",
font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM

plot(gam11.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of FLUORESCENCE

plot(gam11.1, se=T, select=4, shade=T,scale=0,rug=T, xlab="cumul_fluo_10_100")

abline(h=0,lty=2,col=2,lwd=2)

# ----

# ---- SHANNON ~ POS + ST_Z + TIME + TMLD ----

# ---- GAM 12 = Complete data ----

gam12 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(tmld, k=3), data=data)

summary(gam12)

# PLOT GAM 12

```

To display the plots on the same sheet

```
par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

Map from the model

```
myvis.gam(gam12,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

overlay map on map's model

```
par(new=T)
```

balearic island map with transparent background

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

Overly sampling LAT-LON

```
points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)
```

partial effect of ST_DEPTH

```
plot(gam12, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="COMPLETE", font.main=2, col.main="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

partial effect of TIMENORM

```
plot(gam12, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

partial effect of TEMPERATURE

```
plot(gam12, se=T, select=4, shade=T,scale=0,rug=T, xlab="tmld")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

---- GAM 12.1 = Ictioplankton data ----

```
gam12.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
```

```
s(tmld, k=3),
```

```
data=data)
```

```
summary(gam12.1)
```

PLOT GAM 12.1

```

# To display the plots on the same sheet

par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))

# Map from the model

myvis.gam(gam12.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on map's model

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling LAT-LON

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

plot(gam12.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO",
font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM

plot(gam12.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of FLUORESCENCE

plot(gam12.1, se=T, select=4, shade=T,scale=0,rug=T, xlab="tmld")

abline(h=0,lty=2,col=2,lwd=2)

# ----

# ---- SHANNON ~ POS + ST_Z + TIME + SMLD ----

```

---- GAM 13 = Complete data ----

```
gam13 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(smlld, k=3), data=data)

summary(gam13)
```

PLOT GAM 13

```
# To display the plots on the same sheet
par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))

# Map from the model
myvis.gam(gam13,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on map's model
par(new=T)

# balearic island map with transparent background
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)
box()

# Overly sampling LAT-LON
points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH
plot(gam13, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth",
main="COMPLETE", font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM
plot(gam13, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")
abline(h=0,lty=2,col=2,lwd=2)

# partial effect of FLUORESCENCE
plot(gam13, se=T, select=4, shade=T,scale=0,rug=T, xlab="smlld")

abline(h=0,lty=2,col=2,lwd=2)
```

---- GAM 13.1 = Ictioplankton data ----

```
gam13.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
  s(smlld, k=3),
  data=data)

summary(gam13.1)
```

PLOT GAM 13.1

```
# To display the plots on the same sheet
par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))

# Map from the model
myvis.gam(gam13.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on map's model
par(new=T)

# balearic island map with transparent background
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)
box()

# Overly sampling LAT-LON
points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH
plot(gam13.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO",
font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM
plot(gam13.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of FLUORESCENCE
plot(gam13.1, se=T, select=4, shade=T,scale=0,rug=T, xlab="smlld")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# ----
```

```
# ---- SHANNON ~ POS + ST_Z + TIME + FLUO + TMLD ----
```

```
# ---- GAM 14 = Complete data ----
```

```
gam14 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(cumul_fluor_10_100, k=3) + s(tmld, k=3),
data=data)

summary(gam14)
```

```
# PLOT GAM 14
```

```
# To display the plots on the same sheet
```

```
par(mfrow=c(2,3),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

```
# Map from the model
```

```
myvis.gam(gam14,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

```
# overlay map on map's model
```

```
par(new=T)
```

```
# balearic island map with transparent background
```

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

```
# Overly sampling LAT-LON
```

```
points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)
```

```
# partial effect of ST_DEPTH
```

```
plot(gam14, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth",
main="COMPLETE", font.main=2, col.main="red")
```

```

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM

plot(gam14, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of FLUORESCENCE

plot(gam14, se=T, select=4, shade=T,scale=0,rug=T, xlab="cumul_fluo_10_100")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TEMPERATURE

plot(gam14, se=T, select=5, shade=T,scale=0,rug=T, xlab="tmlld")

abline(h=0,lty=2,col=2,lwd=2)

# ---- GAM 14.1 = Ictioplankton data ----

gam14.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
  s(cumul_fluor_10_100, k=3) + s(tmlld, k=3),
  data=data)

summary(gam14.1)

# PLOT GAM 14.1

# To display the plots on the same sheet

par(mfrow=c(2,3),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))

# Map from the model

myvis.gam(gam14.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on map's model

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling LAT-LON

```

```

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

plot(gam14.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO",
font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM

plot(gam14.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of FLUORESCENCE

plot(gam14.1, se=T, select=4, shade=T,scale=0,rug=T, xlab="cumul_fluo_10_100")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TEMPERATURE

plot(gam14.1, se=T, select=5, shade=T,scale=0,rug=T, xlab="tmld")

abline(h=0,lty=2,col=2,lwd=2)

# ----

# ---- SHANNON ~ POS + ST_Z + TIME + FLUO + SMLD ----

# ---- GAM 15 = Complete data ----

gam15 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(cumul_fluor_10_100, k=3) + s(smld, k=3),
data=data)

summary(gam15)

# PLOT GAM 15

# To display the plots on the same sheet

par(mfrow=c(2,3),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))

```

Map from the model

```
myvis.gam(gam15,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

overlay map on map's model

```
par(new=T)
```

balearic island map with transparent background

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)
```

```
box()
```

Overly sampling LAT-LON

```
points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)
```

partial effect of ST_DEPTH

```
plot(gam15, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="COMPLETE", font.main=2, col.main="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

partial effect of TIMENORM

```
plot(gam15, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

partial effect of FLUORESCENCE

```
plot(gam15, se=T, select=4, shade=T,scale=0,rug=T, xlab="cumul_fluo_10_100")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

partial effect of SALINITY

```
plot(gam15, se=T, select=5, shade=T,scale=0,rug=T, xlab="smld")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

---- GAM 15.1 = Ictioplankton data ----

```
gam15.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +  
s(cumul_fluor_10_100, k=3) + s(smld, k=3),  
data=data)  
summary(gam15.1)
```

PLOT GAM 15.1

```

# To display the plots on the same sheet

par(mfrow=c(2,3),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))

# Map from the model

myvis.gam(gam15.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on map's model

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling LAT-LON

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

plot(gam15.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO",
font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM

plot(gam15.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of FLUORESCENCE

plot(gam15.1, se=T, select=4, shade=T,scale=0,rug=T, xlab="cumul_fluo_10_100")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of SALINITY

plot(gam15.1, se=T, select=5, shade=T,scale=0,rug=T, xlab="smlld")

abline(h=0,lty=2,col=2,lwd=2)

```

---- SHANNON ~ POS + ST_Z + TIME + TMID + SMLD ----

---- GAM 16 = Complete data ----

```
gam16 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
           s(tmld, k=3) + s(smld, k=3), data=data)

summary(gam16)
```

PLOT GAM 16

To display the plots on the same sheet

```
par(mfrow=c(2,3),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

Map from the model

```
myvis.gam(gam16,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

overlay map on map's model

```
par(new=T)
```

balearic island map with transparent background

```
plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)
box()
```

Overly sampling LAT-LON

```
points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)
```

partial effect of ST_DEPTH

```
plot(gam16, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth",
main="COMPLETE", font.main=2, col.main="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

partial effect of TIMENORM

```

plot(gam16, se=T, select=3, shade=T, scale=0, rug=T, xlab="timenorm")

abline(h=0, lty=2, col=2, lwd=2)

# partial effect of TEMPERATURE

plot(gam16, se=T, select=4, shade=T, scale=0, rug=T, xlab="tmld")

abline(h=0, lty=2, col=2, lwd=2)

# partial effect of SALINITY

plot(gam16, se=T, select=5, shade=T, scale=0, rug=T, xlab="smld")

abline(h=0, lty=2, col=2, lwd=2)

# ---- GAM 16.1 = Ictioplankton data ----

gam16.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc', k=4) +
  s(tmld, k=3) + s(smld, k=3),
  data=data)

summary(gam16.1)

# PLOT GAM 16.1

# To display the plots on the same sheet

par(mfrow=c(2,3), omi=c(0.2, 0.2, 0.2, 0.2), mai=c(0.7, 0.7, 0.2, 0.2))

# Map from the model

myvis.gam(gam16.1, view=c('lon', 'lat'), plot.type="contour", color="jet", too.far=0.05)

# overlay map on map's model

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0, alpha=0.05)

box()

# Overly sampling LAT-LON

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

```

```
plot(gam16.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO",
font.main=2, col.main="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of TIMENORM
```

```
plot(gam16.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of TEMPERATURE
```

```
plot(gam16.1, se=T, select=4, shade=T,scale=0,rug=T, xlab="tmld")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of SALINITY
```

```
plot(gam16.1, se=T, select=5, shade=T,scale=0,rug=T, xlab="smlld")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# ----
```

```
# ---- SHANNON ~ POS + ST_Z + TIME + FLUO + TMLD + SMLD ----
```

```
# ---- GAM 17 = Complete data ----
```

```
gam17 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
```

```
s(cumul_fluor_10_100, k=3) + s(tmld, k=3) + s(smlld, k=3),
```

```
data=data)
```

```
summary(gam17)
```

```
# PLOT GAM 17
```

```
# To display the plots on the same sheet
```

```
par(mfrow=c(2,3),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

```
# Map from the model
```

```
myvis.gam(gam17,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)
```

overlay map on map's model

```

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling LAT-LON

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

plot(gam17, se=T, select=2, shade=T, scale=0, rug=T, xlab="log st_depth",
main="COMPLETE", font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM

plot(gam17, se=T, select=3, shade=T, scale=0, rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of FLUORESCENCE

plot(gam17, se=T, select=4, shade=T, scale=0, rug=T, xlab="cumul_fluo_10_100")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TEMPERATURE

plot(gam17, se=T, select=5, shade=T, scale=0, rug=T, xlab="tmld")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of SALINITY

plot(gam17, se=T, select=6, shade=T, scale=0, rug=T, xlab="smld")

abline(h=0,lty=2,col=2,lwd=2)

# ---- GAM 17.1 = Ictioplankton data ----

gam17.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(cumul_fluor_10_100, k=3) + s(tmld, k=3) + s(smld, k=3),
data=data)

summary(gam17.1)

```

PLOT GAM 17.1

```

# To display the plots on the same sheet

par(mfrow=c(2,3),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))

# Map from the model

myvis.gam(gam17.1,view=c('lon','lat'),plot.type="contour",color="jet",too.far=0.05)

# overlay map on map's model

par(new=T)

# balearic island map with transparent background

plot(spain, xlim=xlim, ylim=ylim, col="grey40", bg="#FFFFFF0D") # rgb=c(0,0,0,alpha=0.05)

box()

# Overly sampling LAT-LON

points(data$lon, data$lat, pch=20, col=makeTransparent("black"), cex=.3)

# partial effect of ST_DEPTH

plot(gam17.1, se=T, select=2, shade=T,scale=0,rug=T, xlab="log st_depth", main="ICTIO",
font.main=2, col.main="red")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TIMENORM

plot(gam17.1, se=T, select=3, shade=T,scale=0,rug=T, xlab="timenorm")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of FLUORESCENCE

plot(gam17.1, se=T, select=4, shade=T,scale=0,rug=T, xlab="cumul_fluo_10_100")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of TEMPERATURE

plot(gam17.1, se=T, select=5, shade=T,scale=0,rug=T, xlab="tmld")

abline(h=0,lty=2,col=2,lwd=2)

# partial effect of SALINITY

plot(gam17.1, se=T, select=6, shade=T,scale=0,rug=T, xlab="smld")

```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# -----
```

```
# ---- GAM SIGNIFICANT COVARIATES ----
```

```
# -----
```

```
# GAM with all geo+hydro covariates for complete data set
```

```
# only ST_Z, TIME and SMLD are significant
```

```
# ---- SHANNON ~ ST_Z + TIME + SMLD ----
```

```
gam99 <- gam(Shannon ~ s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
```

```
factor(year),
```

```
data=data)
```

```
summary(gam99)
```

```
AIC(gam99)
```

```
# PLOT GAM 99
```

```
# To display the plots on the same sheet
```

```
par(mfrow=c(2,2),omi=c(0.2,0.2,0.2,0.2),mai=c(0.7,0.7,0.2,0.2))
```

```
# partial effect of ST_DEPTH
```

```
plot(gam99, se=T, select=1, shade=T,scale=0,rug=T, xlab="log st_depth",  
main="COMPLETE", font.main=2, col.main="red")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of TIMENORM
```

```
plot(gam99, se=T, select=2, shade=T,scale=0,rug=T, xlab="timenorm")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
# partial effect of SALINITY
```

```
plot(gam99, se=T, select=3, shade=T,scale=0,rug=T, xlab="smld")
```

```
abline(h=0,lty=2,col=2,lwd=2)
```

```
pred.stz <- predict(gam99, newdata=data, type="response")
```

```
plot(data$Shannon ~ log(data$st_depth))
```

```
lines(pred.stz, add=TRUE, col="red")
```

```
pred.time <- predict(gam99, newdata=data, type="response")
```

```
plot(data$Shannon ~ data$timenorm)
```

```
lines(pred.time, add=T)
```

```
# -----
```

```
# ---- COMPARING MODELS ----
```

```
# -----
```

```
# F-TEST --> H0: the simpler model is the correct
```

```
# p-value > 0.05 --> H0 is correct
```

```
# ----
```

```
# ---- POS+ST_Z+TIME+FLUO+SAL VS POS+ST_Z+TIME+FLUO+TEMP+SAL ----
```

```
# ---- Ictioplankton data 15.1 VS 17.1 ----
```

```
anova(gam15.1,gam17.1,test='F')
```

```
# ----
```

```
# ---- POS+ST_Z+TIME+FLUO+TEMP VS POS+ST_Z+TIME+FLUO+TEMP+SAL ----
```

---- Ictioplankton data 14.1 VS 17.1 ----

```
anova(gam14.1,gam17.1,test='F')
```

---- GEO+FLUO+SAL VS GEO+FLUO+TEMP+SAL ----

---- Complete data ----

```
gam15 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(cumul_fluor_10_100, k=3) + s(smlld, k=3), data=data)
```

```
gam17 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(cumul_fluor_10_100, k=3) + s(tmld, k=3) + s(smlld, k=3), data=data)
```

```
anova(gam15,gam17,test='F')
```

---- Ictioplankton data ----

```
gam15.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(cumul_fluor_10_100, k=3) + s(smlld, k=3), data=data)
```

```
gam17.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(cumul_fluor_10_100, k=3) + s(tmld, k=3) + s(smlld, k=3), data=data)
```

```
anova(gam15.1,gam17.1,test='F')
```

---- GEO+FLUO VS GEO+FLUO+SAL ----

---- Complete data ----

```
gam11 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
s(cumul_fluor_10_100, k=3), data=data)
```

```
gam15 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
```

s(cumul_fluor_10_100, k=3) + s(smld, k=3), data=data)

anova(gam11,gam15,test='F')

---- Ictioplankton data ----

gam11.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
 s(cumul_fluor_10_100, k=3), data=data)

gam15.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
 s(cumul_fluor_10_100, k=3) + s(smld, k=3), data=data)

anova(gam11.1,gam15.1,test='F')

---- GEO+SAL VS GEO+FLUO+SAL ----

---- Complete data ----

gam13 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
 s(smld, k=3), data=data)

gam15 <- gam(Shannon ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
 s(cumul_fluor_10_100, k=3) + s(smld, k=3), data=data)

anova(gam13,gam15,test='F')

---- Ictioplankton data ----

gam13.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
 s(smld, k=3), data=data)

gam15.1 <- gam(Shan1 ~ s(lat, lon, k=15) + s(log(st_depth), k=3) + s(timenorm, bs='cc',k=4) +
 s(cumul_fluor_10_100, k=3) + s(smld, k=3), data=data)



Gobierno
de España





Fundación Biodiversidad



Unión Europea

Fondo Europeo Marítimo y de Pesca (FEMP)



```
anova(gam13.1,gam15.1,test='F')
```

8. DIV_MAPS.R:visor

---- DIVERPEL: MAPPING BIODIVERSITY INDEXES ----

Author: ESTHER BARBER

Created on: JAN 2020

Last modification : 2020-02-11

---- PACKAGES ----

library(tidyverse)

library(xlsx)

library(dplyr)

library(readxl)

library(reshape2)

Packages MAPS

library(MBA)

library(colorRamps)

library(maps)

library(maptools)

library(raster)

library(ggrepel) # geom_text_repel function

Calling function defined in R

#source("C:/Users/esther.barber/Desktop/R software/LM and GLM summary export.R")

---- DIVERPEL DATA ----

```

data
read_excel("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/DIVER_excel/07.GA
M_data.xlsx") <-
data <- data.frame(data)
str(data)
data <- data %>%
  mutate_at(vars(operation, survey, estacion, year), funs(factor))
# _____

```

---- GENERAL MAP ----

```

# Load maps data:
spain <- getData("GADM", country = "ESP", level = 0)
# Map limits
xlim <- c(min(data$lon, na.rm=F), max(data$lon, na.rm=F))
ylim <- c(min(data$lat, na.rm=F), max(data$lat, na.rm=F))
# ----
# _____

```

---- COMPLETE DATA TO DISPLAY ON MAPS ----

```

# data I want to plot
dat_comp <- data.frame(operation = data$operation,
                        lat=data$lat, lon=data$lon, year=data$year,
                        Shannon=data$Shannon, SpsNum=data$SpsNum,
                        X1.Simps=data$X1.Simpson, inv_Simps=data$inv_Simps)
# ----

```

---- COMPLETE SHANNON INDEX SPATIAL DISTRIBUTION 2012-2016 ----

```

mba <- mba.surf(dat_comp[ , c('lon', 'lat', 'Shannon')], 200, 200)

dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)

df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'Shannon')

Shannon_imp <- dat_comp[!(dat_comp$Shannon <= 2),, drop=FALSE]

=
```

```

png(filename
paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM
RESULTS/MAPS/COMPLETE/COM_Shannon_548_12-16_200224.png"),
width = 20, height = 20, units = "cm", res = 400)
```

```

p <- ggplot() +
  geom_raster(data = df, aes(lon, lat, fill = Shannon), interpolate = T) +
  geom_contour(data = df, aes(lon, lat, z = Shannon)) +
  geom_point(data = data, aes(lon, lat, colour=year), shape=1, size = 3) +
  geom_line(data=data, aes(lon, lat, z=37.5)) +
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$Shannon))) +
  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
color="grey40") +
  coord_equal(xlim = xlim, ylim = ylim) + # coord_cartesian saca mapa mÁs quadradito
  ggtitle(paste0("Shannon 2012-2016 complete")) +
  # geom_label_repel(data=Shannon_imp, aes(x=lon, y=lat, label=operation), colour="black",
size=3.5) +
  theme_bw() +
  theme(panel.background = element_rect(fill = "white"),
  panel.grid = element_blank(),
  plot.title = element_text(hjust=0.5),

```

```

axis.title= element_blank(),

legend.title=element_blank()

print(p)

dev.off()

# ---- COMPLETE SHANNON INDEX SPATIAL DISTRIBUTION BY YEAR ----

myfunction2 <- function(x){

  for(i in 2012:2016){

    temp <- x %>%
      filter(year == i)

    # mba.surf returns a surface approximated from a bivariate scatter of data points using
    # multilevel B-splines

    mba <- mba.surf(temp[ , c('lon', 'lat', 'Shannon')], 200, 200)

    dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)

    df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'Shannon')

    Shannon_imp <- temp[!(temp$Shannon <= 2),, drop=FALSE]

    p <- ggplot() +


      geom_raster(data = df, aes(lon, lat, fill = Shannon), interpolate = T) +


      geom_contour(data = df, aes(lon, lat, z = Shannon)) +


      geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +


      scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$Shannon))) +


      geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
      color="grey40") +


      coord_equal(xlim = xlim, ylim = ylim) +


      geom_label_repel(data=Shannon_imp, aes(x=lon, y=lat, label=operation), colour="black",
      size=3.5) +


      ggtitle(paste0("Shannon ",i, " complete")) +


      theme_bw() +
  }
}

```

```
theme(panel.background = element_rect(fill = "white"),
```

```
  panel.grid = element_blank(),
```

```
  plot.title = element_text(hjust=0.5),
```

```
  axis.title= element_blank(),
```

```
  legend.title=element_blank())
```

```
  png(filename
```

```
=
```

```
  paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  
RESULTS/MAPS/COMPLETE/COM_Shannon_548_",".i,"_200224.png"),
```

```
  width = 20, height = 15, units = "cm", res = 400)
```

```
  print(p)
```

```
  dev.off()
```

```
}
```

```
}
```

```
myfunction2(dat_comp)
```

```
# ----
```

```
# ---- COMPLETE NUM.SPS INDEX SPATIAL DISTRIBUTION 2012-2016 ----
```

```
mba <- mba.surf(dat_comp[, c('lon', 'lat', 'SpsNum')], 200, 200)
```

```
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
```

```
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'SpsNum')
```

```
df$value[df$SpsNum < 0] <- 0
```

```
  png(filename
```

```
=
```

```
  paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  
RESULTS/MAPS/COMPLETE/COM_Num.Sps_548_12-16_200224.png"),
```

```
  width = 20, height = 20, units = "cm", res = 400)
```

```

p <- ggplot() +  

  geom_raster(data = df, aes(lon, lat, fill = SpsNum), interpolate = T) +  

  geom_contour(data = df, aes(lon, lat, z = SpsNum)) +  

  geom_point(data = data, aes(lon, lat, colour=year), shape=1, size = 3) +  

  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$SpsNum))) +  

  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",  

color="grey40") +  

  coord_equal(xlim = xlim, ylim = ylim) +  

  ggtitle(paste0("SpsNum 2012-2016 complete")) +  

  theme_bw() +  

  theme(panel.background = element_rect(fill = "white"),  

        panel.grid = element_blank(),  

        plot.title = element_text(hjust=0.5),  

        axis.title= element_blank(),  

        legend.title=element_blank())  

print(p)  

dev.off()

```

---- COMPLETE NUM.SPS INDEX SPATIAL DISTRIBUTION BY YEAR ----

```

myfunction2 <- function(x){  

  for(i in 2012:2016){  

    temp <- x %>%  

      filter(year == i)  

    # mba.surf returns a surface approximated from a bivariate scatter of data points using  

    # multilevel B-splines  

    mba <- mba.surf(temp[ , c('lon', 'lat', 'SpsNum')], 200, 200)  

    dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
}

```

```

df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'SpsNum')

df$value[df$SpsNum < 0] <- 0

p <- ggplot() +  

  geom_raster(data = df, aes(lon, lat, fill = SpsNum), interpolate = T) +  

  geom_contour(data = df, aes(lon, lat, z = SpsNum)) +  

  geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +  

  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$SpsNum))) + # all  

maps same color scale  

  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",  

color="grey40") +  

  coord_equal(xlim = xlim, ylim = ylim) +  

  ggtitle(paste0("SpsNum ",i, " complete")) +  

  theme_bw() +  

  theme(panel.background = element_rect(fill = "white"),  

  panel.grid = element_blank(),  

  plot.title = element_text(hjust=0.5),  

  axis.title= element_blank(),  

  legend.title=element_blank())

  png(filename  

  paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  

RESULTS/MAPS/COMPLETE/COM_SpsNum_548_",i,"_200224.png"),  

  width = 20, height = 15, units = "cm", res = 400)  

  print(p)  

  dev.off()  

}

}

myfunction2(dat_comp)

```

---- COMPLETE (1-SIMPSON) INDEX SPATIAL DISTRIBUTION 2012-2016 ----

```

mba <- mba.surf(dat_comp[, c('lon', 'lat', 'X1.Simps')], 200, 200)

dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)

df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'X1.Simps')

df$value[df$X1.Simps < 0] <- 0

=
```

```

png(filename
paste0("C:/Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM
RESULTS/MAPS/COMPLETE/COM_1-Simpson_549_12-16.png"),
width = 20, height = 20, units = "cm", res = 400)
```

```

p <- ggplot() +
  geom_raster(data = df, aes(lon, lat, fill = X1.Simps), interpolate = T) +
  geom_contour(data = df, aes(lon, lat, z = X1.Simps)) +
  geom_point(data = data, aes(lon, lat, colour=year), shape=1, size = 3) +
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$X1.Simps))) +
  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
color="grey40") +
  coord_equal(xlim = xlim, ylim = ylim) +
  ggtitle(paste0("1-Simpson 2012-2016 complete")) +
  theme_bw() +
  theme(panel.background = element_rect(fill = "white"),
  panel.grid = element_blank(),
  plot.title = element_text(hjust=0.5),
  axis.title= element_blank(),
  legend.title=element_blank())

```

```
print(p)
```

```
dev.off()
```

```
# ---- COMPLETE (1-SIMPSON) INDEX SPATIAL DISTRIBUTION BY YEAR ----
```

```
myfunction2 <- function(x){  
  
  for(i in 2012:2016){  
  
    temp <- x %>%  
  
    filter(year == i)  
  
    # mba.surf returns a surface approximated from a bivariate scatter of data points using  
    # multilevel B-splines  
  
    mba <- mba.surf(temp[ , c('lon', 'lat', 'X1.Simps')], 200, 200)  
  
    dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)  
  
    df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'X1.Simps')  
  
    df$value[df$X1.Simps < 0] <- 0  
  
  
    p <- ggplot() +  
      geom_raster(data = df, aes(lon, lat, fill = X1.Simps), interpolate = T) +  
      geom_contour(data = df, aes(lon, lat, z = X1.Simps)) +  
      geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +  
      scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$X1.Simps))) +  
      geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",  
      color="grey40") +  
      coord_equal(xlim = xlim, ylim = ylim) +  
      ggtitle(paste0("1-Simpson ", i, " complete")) +  
      theme_bw() +  
      theme(panel.background = element_rect(fill = "white"),  
            panel.grid = element_blank(),
```

```
plot.title = element_text(hjust=0.5),
```

```
axis.title= element_blank(),
```

```
legend.title=element_blank())
```

```
png(filename  
paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  
RESULTS/MAPS/COMPLETE/COM_1-Simpson_549_","i,"_jan20.png"),
```

```
width = 20, height = 15, units = "cm", res = 400)
```

```
print(p)
```

```
dev.off()
```

```
}
```

```
}
```

```
myfunction2(dat_comp)
```

```
# ----
```

```
# ---- COMPLETE (1/SIMPSON) INDEX SPATIAL DISTRIBUTION 2012-2016 ----
```

```
mba <- mba.surf(dat_comp[, c('lon', 'lat', 'inv_Simps')], 200, 200)
```

```
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
```

```
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'inv_Simps')
```

```
png(filename  
paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  
RESULTS/MAPS/COMPLETE/COM_inv_Simpson_549_12-16.png"),
```

```
width = 20, height = 20, units = "cm", res = 400)
```

```
p <- ggplot() +
```

```
geom_raster(data = df, aes(lon, lat, fill = inv_Simps), interpolate = T) +
```

```

geom_contour(data = df, aes(lon, lat, z = inv_Simps)) +
  geom_point(data = data, aes(lon, lat, colour = year), shape=1, size = 3) +
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(1, max(data$inv_Simps))) + # el min
  de invSimps es siempre 1

  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
  color="grey40") +
  coord_equal(xlim = xlim, ylim = ylim) +
  ggtitle(paste0("1/Simpson 2012-2016 complete")) +
  theme_bw() +
  theme(panel.background = element_rect(fill = "white"),
  panel.grid = element_blank(),
  plot.title = element_text(hjust=0.5),
  axis.title= element_blank(),
  legend.title=element_blank())

print(p)
dev.off()

```

---- COMPLETE (1/SIMPSON) INDEX SPATIAL DISTRIBUTION BY YEAR ----

```

myfunction2 <- function(x){

  for(i in 2012:2016){

    temp <- x %>%
      filter(year == i)

    # mba.surf returns a surface approximated from a bivariate scatter of data points using
    # multilevel B-splines

    mba <- mba.surf(temp[ , c('lon', 'lat', 'inv_Simps')], 200, 200)

    dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)

    df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'inv_Simps')
  }
}

```

```

p <- ggplot() +  

  geom_raster(data = df, aes(lon, lat, fill = inv_Simps), interpolate = T) +  

  geom_contour(data = df, aes(lon, lat, z = inv_Simps)) +  

  geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +  

  scale_fill_gradientn(colours = matlab.like2(7), limits = c(1, max(data$inv_Simps))) +  

  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",  

  color="grey40") +  

  coord_equal(xlim = xlim, ylim = ylim) +  

  ggtitle(paste0("1/Simpson ", i, " complete")) +  

  theme_bw() +  

  theme(panel.background = element_rect(fill = "white"),  

        panel.grid = element_blank(),  

        plot.title = element_text(hjust=0.5),  

        axis.title= element_blank(),  

        legend.title=element_blank())  

  png(filename  

  paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  

RESULTS/MAPS/COMPLETE/COM_inv_Simpson_549_", i, "_jan20.png"),  

width = 20, height = 15, units = "cm", res = 400)  

  print(p)  

  dev.off()  

}  

}
  

myfunction2(dat_comp)  

# ----  

# _____

```

---- ICTIOPLANKTON DATA TO DISPLAY ON MAPS ----

```
# data I want to plot

dat_ict <- data.frame(operation = data$operation,
                       lat=data$lat, lon=data$lon, year=data$year,
                       Shannon=data$Shan1, SpsNum=data$SpsNum1,
                       X1.Simps=data$X1.Simp1, inv_Simps=data$inv_Simp1)

# ----
```

---- ICTIOPLANKTON SHANNON INDEX SPATIAL DISTRIBUTION 2012-2016 ----

```
mba <- mba.surf(dat_ict[ , c('lon', 'lat', 'Shannon')], 200, 200)

dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)

df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'Shannon')

Shannon_imp <- dat_ict[!(dat_ict$Shannon <= 2),, drop=FALSE]
```

```
png(filename = paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  
RESULTS/MAPS/ICTIOPLANKTON/ICT_Shannon_548_200224.png"),
     width = 20, height = 20, units = "cm", res = 400)
```

```
p <- ggplot() +
  geom_raster(data = df, aes(lon, lat, fill = Shannon), interpolate = T) +
  geom_contour(data = df, aes(lon, lat, z = Shannon)) +
  geom_point(data = data, aes(lon, lat, colour=year), shape=1, size = 3) +
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$Shannon))) +
  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
               color="grey40") +
  coord_equal(xlim = xlim, ylim = ylim) + # coord_cartesian saca mapa mÁ;s quadradito
  ggtitle(paste0("Shannon 2012-2016 ICTIOPLANKTON")) +
```

```
geom_label_repel(data=Shannon_imp, aes(x=lon, y=lat, label=operation), colour="black",
size=3.5) +
```

```
theme_bw() +
```

```
theme(panel.background = element_rect(fill = "white"),
```

```
panel.grid = element_blank(),
```

```
plot.title = element_text(hjust=0.5),
```

```
axis.title= element_blank(),
```

```
legend.title=element_blank())
```

```
print(p)
```

```
dev.off()
```

```
# ---- ICTIOPLANKTON SHANNON INDEX SPATIAL DISTRIBUTION BY YEAR ----
```

```
myfunction2 <- function(x){
```

```
for(i in 2012:2016){
```

```
temp <- x %>%
```

```
filter(year == i)
```

```
# mba.surf returns a surface approximated from a bivariate scatter of data points using
multilevel B-splines
```

```
mba <- mba.surf(temp[, c('lon', 'lat', 'Shannon')], 200, 200)
```

```
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
```

```
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'Shannon')
```

```
Shannon_imp <- temp[!(temp$Shannon <= 2.0),, drop=FALSE]
```

```
p <- ggplot() +
```

```
geom_raster(data = df, aes(lon, lat, fill = Shannon), interpolate = T) +
```

```
geom_contour(data = df, aes(lon, lat, z = Shannon)) +
```

```
geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +
```

```

scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$Shannon))) +  

  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",  

color="grey40") +  

  coord_equal(xlim = xlim, ylim = ylim) +  

  ggtitle(paste0("Shannon ",i, " ICTIOPLANKTON")) +  

  geom_label_repel(data=Shannon_imp, aes(x=lon, y=lat, label=operation), colour="black",  

size=3.5) +  

  theme_bw() +  

  theme(panel.background = element_rect(fill = "white"),  

panel.grid = element_blank(),  

plot.title = element_text(hjust=0.5),  

axis.title= element_blank(),  

legend.title=element_blank())  
  

  png(filename  

paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  

RESULTS/MAPS/ICTIOPLANKTON/ICT_Shannon_548_",i,"_feb20_200224.png"),  

width = 20, height = 15, units = "cm", res = 400)  

print(p)  

dev.off()  

}  

}  
  

myfunction2(dat_ict)  
  

# ----  
  

# ---- ICTIOPLANKTON NUM.SPS INDEX SPATIAL DISTRIBUTION 2012-2016 ----  
  

mba <- mba.surf(dat_ict[ , c('lon', 'lat', 'SpsNum')], 200, 200)

```

```

dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)

df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'SpsNum')

df$value[df$SpsNum < 0] <- 0

=
```

```

png(filename
paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM
RESULTS/MAPS/ICTIOPLANKTON/ICT_Num.Sps_548_12-16_200224.png"),
width = 20, height = 20, units = "cm", res = 400)

p <- ggplot() +
  geom_raster(data = df, aes(lon, lat, fill = SpsNum), interpolate = T) +
  geom_contour(data = df, aes(lon, lat, z = SpsNum)) +
  geom_point(data = data, aes(lon, lat, colour=year), shape=1, size = 3) +
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$SpsNum))) +
  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
color="grey40") +
  coord_equal(xlim = xlim, ylim = ylim) +
  ggtitle(paste0("SpsNum 2012-2016 ICTIOPLANKTON")) +
  theme_bw() +
  theme(panel.background = element_rect(fill = "white"),
        panel.grid = element_blank(),
        plot.title = element_text(hjust=0.5),
        axis.title= element_blank(),
        legend.title=element_blank())
  print(p)
  dev.off()

# ---- ICTIOPLANKTON NUM.SPS INDEX SPATIAL DISTRIBUTION BY YEAR ----

```

```
myfunction2 <- function(x){
```

```
for(i in 2012:2016){
```

```
  temp <- x %>%
```

```
    filter(year == i)
```

```
# mba.surf returns a surface approximated from a bivariate scatter of data points using
multilevel B-splines
```

```
mba <- mba.surf(temp[ , c('lon', 'lat', 'SpsNum')], 200, 200)
```

```
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
```

```
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'SpsNum')
```

```
df$value[df$SpsNum < 0] <- 0
```

```
p <- ggplot() +
```

```
  geom_raster(data = df, aes(lon, lat, fill = SpsNum), interpolate = T) +
```

```
  geom_contour(data = df, aes(lon, lat, z = SpsNum)) +
```

```
  geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +
```

```
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$SpsNum))) + # all
maps same color scale
```

```
  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
color="grey40") +
```

```
  coord_equal(xlim = xlim, ylim = ylim) +
```

```
  ggtitle(paste0("SpsNum ", i, " ICTIOPLANKTON")) +
```

```
  theme_bw() +
```

```
  theme(panel.background = element_rect(fill = "white"),
```

```
    panel.grid = element_blank(),
```

```
    plot.title = element_text(hjust=0.5),
```

```
    axis.title= element_blank(),
```

```
    legend.title=element_blank())
```

```
=  

png(filename  

paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  

RESULTS/MAPS/ICTIOPLANKTON/ICT_SpsNum_548_", i, "_200224.png"),
```

```
width = 20, height = 15, units = "cm", res = 400)
```

```
print(p)
```

```
dev.off()
```

```
}
```

```
}
```

```
myfunction2(dat_ict)
```

```
# ----
```

```
# ---- ICTIOPLANKTON (1-SIMPSON) INDEX SPATIAL DISTRIBUTION 2012-2016 ----
```

```
mba <- mba.surf(dat_ict[ , c('lon', 'lat', 'X1.Simps')], 200, 200)
```

```
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
```

```
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'X1.Simps')
```

```
df$value[df$X1.Simps < 0] <- 0
```

```
png(filename
```

```
=
```

```
paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  
RESULTS/MAPS/ICTIOPLANKTON/COM_1-Simpson_549_12-16.png"),
```

```
width = 20, height = 20, units = "cm", res = 400)
```

```
p <- ggplot() +
```

```
geom_raster(data = df, aes(lon, lat, fill = X1.Simps), interpolate = T) +
```

```
geom_contour(data = df, aes(lon, lat, z = X1.Simps)) +
```

```
geom_point(data = data, aes(lon, lat, colour=year), shape=1, size = 3) +
```

```
scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$X1.Simp1))) +
```

```
geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",  
color="grey40") +
```

```
coord_equal(xlim = xlim, ylim = ylim) +
```

```
ggtitle(paste0("1-Simpson 2012-2016 ICTIOPLANKTON")) +
```

```

theme_bw() +  

theme(panel.background = element_rect(fill = "white"),  

  panel.grid = element_blank(),  

  plot.title = element_text(hjust=0.5),  

  axis.title= element_blank(),  

  legend.title=element_blank())  

print(p)  

dev.off()  

# ---- ICTIOPLANKTON (1-SIMPSON) INDEX SPATIAL DISTRIBUTION BY YEAR ----  

myfunction2 <- function(x){  

  for(i in 2012:2016){  

    temp <- x %>%  

    filter(year == i)  

    # mba.surf returns a surface approximated from a bivariate scatter of data points using  

    # multilevel B-splines  

    mba <- mba.surf(temp[ , c('lon', 'lat', 'X1.Simps')], 200, 200)  

    dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)  

    df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'X1.Simps')  

    df$value[df$X1.Simps < 0] <- 0  

  p <- ggplot() +  

  geom_raster(data = df, aes(lon, lat, fill = X1.Simps), interpolate = T) +  

  geom_contour(data = df, aes(lon, lat, z = X1.Simps)) +  

  geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +  

  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(data$X1.Simp1))) +

```

```
geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
color="grey40") +
```

```
coord_equal(xlim = xlim, ylim = ylim) +
ggtitle(paste0("1-Simpson ",i, " ICTIOPLANKTON")) +
theme_bw() +
theme(panel.background = element_rect(fill = "white"),
panel.grid = element_blank(),
plot.title = element_text(hjust=0.5),
axis.title= element_blank(),
legend.title=element_blank())
```

```
png(filename =
paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM
RESULTS/MAPS/ICTIOPLANKTON/COM_1-Simpson_549_","_i,_jan20.png"),
width = 20, height = 15, units = "cm", res = 400)
```

```
print(p)
dev.off()
}
}
```

```
myfunction2(dat_ict)
```

```
# ----
```

```
# ---- ICTIOPLANKTON (1/SIMPSON) INDEX SPATIAL DISTRIBUTION 2012-2016 ----
```

```
mba <- mba.surf(dat_ict[ , c('lon', 'lat', 'inv_Simps')], 200, 200)
```

```
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
```

```
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'inv_Simps')
```

png(filename
 paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM
 RESULTS/MAPS/ICTIOPLANKTON/COM_inv_Simpson_549_12-16.png"),
 width = 20, height = 20, units = "cm", res = 400)

```

p <- ggplot() +  

  geom_raster(data = df, aes(lon, lat, fill = inv_Simps), interpolate = T) +  

  geom_contour(data = df, aes(lon, lat, z = inv_Simps)) +  

  geom_point(data = data, aes(lon, lat, colour = year), shape=1, size = 3) +  

  scale_fill_gradientn(colours = matlab.like2(7), limits = c(1, max(data$inv_Simp1))) + # el min  

  de invSimps es siempre 1  

  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",  

  color="grey40") +  

  coord_equal(xlim = xlim, ylim = ylim) +  

  ggtitle(paste0("1/Simpson 2012-2016 ICTIOPLANKTON")) +  

  theme_bw() +  

  theme(panel.background = element_rect(fill = "white"),  

  panel.grid = element_blank(),  

  plot.title = element_text(hjust=0.5),  

  axis.title= element_blank(),  

  legend.title=element_blank())  

print(p)  

dev.off()
  
```

---- ICTIOPLANKTON (1/SIMPSON) INDEX SPATIAL DISTRIBUTION BY YEAR ----

```

myfunction2 <- function(x){  

  for(i in 2012:2016){  

    temp <- x %>%
  }
  
```

filter(year == i)

mba.surf returns a surface approximated from a bivariate scatter of data points using multilevel B-splines

```
mba <- mba.surf(temp[ , c('lon', 'lat', 'inv_Simps')], 200, 200)
```

```
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
```

```
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'inv_Simps')
```

```
p <- ggplot() +
```

```
  geom_raster(data = df, aes(lon, lat, fill = inv_Simps), interpolate = T) +
```

```
  geom_contour(data = df, aes(lon, lat, z = inv_Simps)) +
```

```
  geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +
```

```
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(1, max(data$inv_Simp1))) +
```

```
  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey", color="grey40") +
```

```
  coord_equal(xlim = xlim, ylim = ylim) +
```

```
  ggtitle(paste0("1/Simpson ", i, " ICTIOPLANKTON")) +
```

```
  theme_bw() +
```

```
  theme(panel.background = element_rect(fill = "white"),
```

```
    panel.grid = element_blank(),
```

```
    plot.title = element_text(hjust=0.5),
```

```
    axis.title= element_blank(),
```

```
    legend.title=element_blank())
```

```
png(filename
```

```
=
```

```
paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  
RESULTS/MAPS/ICTIOPLANKTON/COM_inv_Simpson_549_", i, "_jan20.png"),
```

```
width = 20, height = 15, units = "cm", res = 400)
```

```
print(p)
```

```
dev.off()
```

```
}
```



myfunction2(dat_ict)

#

---- TUNA DATA TO DISPLAY ON MAPS ----

```
# data I want to plot
```

```
dat_tun <- data.frame(operation = data$operation,
```

lat=data\$lat, lon=data\$lon, year=data\$year,

Shannon=data\$Shan2, SpsNum=data\$SpsNum2,

```
X1.Simps=data$X1.Simp2, inv_Simps=data$inv_Simp2)
```

---- TUNA SHANNON INDEX SPATIAL DISTRIBUTION 2012-2016 ----

```
mba <- mba.surf(dat_tun[ , c('lon', 'lat', 'Shannon')], 200, 200)
```

```
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
```

```
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'Shannon')
```

```
df$value[df$Shannon < 0] <- 0
```

```
png(filename =  
  paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  
RESULTS/MAPS/TUNA/TUN_Shannon_548_12-16.png"),
```

width = 20, height = 20, units = "cm", res = 400)

```
p <- ggplot() +
```

```
geom_raster(data = df, aes(lon, lat, fill = Shannon), interpolate = T) +
```

```

geom_contour(data = df, aes(lon, lat, z = Shannon)) +
  geom_point(data = data, aes(lon, lat, colour=year), shape=1, size = 3) +
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(dat_tun$Shannon))) +
  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
  color="grey40") +
  coord_equal(xlim = xlim, ylim = ylim) + # coord_cartesian saca mapa mÁs quadradito
  ggtitle(paste0("Shannon 2012-2016 TUNA")) +
  theme_bw() +
  theme(panel.background = element_rect(fill = "white"),
  panel.grid = element_blank(),
  plot.title = element_text(hjust=0.5),
  axis.title= element_blank(),
  legend.title=element_blank())
print(p)
dev.off()

```

---- TUNA SHANNON INDEX SPATIAL DISTRIBUTION BY YEAR ----

```

myfunction2 <- function(x){
  for(i in 2012:2016){
    temp <- x %>%
      filter(year == i)

    # mba.surf returns a surface approximated from a bivariate scatter of data points using
    # multilevel B-splines

    mba <- mba.surf(temp[ , c('lon', 'lat', 'Shannon')], 200, 200)
    dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
    df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'Shannon')
    df$value[df$Shannon < 0] <- 0
  }
}

```

```

p <- ggplot() +  

  geom_raster(data = df, aes(lon, lat, fill = Shannon), interpolate = T) +  

  geom_contour(data = df, aes(lon, lat, z = Shannon)) +  

  geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +  

  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(dat_tun$Shannon))) +  

  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",  

  color="grey40") +  

  coord_equal(xlim = xlim, ylim = ylim) +  

  ggtitle(paste0("Shannon ", i, " TUNA")) +  

  theme_bw() +  

  theme(panel.background = element_rect(fill = "white"),  

        panel.grid = element_blank(),  

        plot.title = element_text(hjust=0.5),  

        axis.title= element_blank(),  

        legend.title=element_blank())  

  png(filename  

  paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  

RESULTS/MAPS/TUNA/TUN_Shannon_548_", i, "_Feb20.png"),  

  width = 20, height = 15, units = "cm", res = 400)  

  print(p)  

  dev.off()  

}  

}

```

myfunction2(dat_tun)

---- TUNA NUM.SPS INDEX SPATIAL DISTRIBUTION 2012-2016 ----

```

mba <- mba.surf(dat_tun[ , c('lon', 'lat', 'SpsNum')], 200, 200)

dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)

df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'SpsNum')

df$value[df$SpsNum < 0] <- 0

=>

png(filename
paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM
RESULTS/MAPS/TUNA/TUN_Num.Sps_548_12-16.png"),
width = 20, height = 20, units = "cm", res = 400)

p <- ggplot() +
  geom_raster(data = df, aes(lon, lat, fill = SpsNum), interpolate = T) +
  geom_contour(data = df, aes(lon, lat, z = SpsNum)) +
  geom_point(data = data, aes(lon, lat, colour=year), shape=1, size = 3) +
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(dat_tun$SpsNum))) +
  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
color="grey40") +
  coord_equal(xlim = xlim, ylim = ylim) +
  ggtitle(paste0("SpsNum 2012-2016 TUNA")) +
  theme_bw() +
  theme(panel.background = element_rect(fill = "white"),
  panel.grid = element_blank(),
  plot.title = element_text(hjust=0.5),
  axis.title= element_blank(),
  legend.title=element_blank())

print(p)

dev.off()

# ---- TUNA NUM.SPS INDEX SPATIAL DISTRIBUTION BY YEAR ----

```

```

myfunction2 <- function(x){

  for(i in 2012:2016){

    temp <- x %>%
      filter(year == i)

    # mba.surf returns a surface approximated from a bivariate scatter of data points using
    # multilevel B-splines

    mba <- mba.surf(temp[ , c('lon', 'lat', 'SpsNum')], 200, 200)

    dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)

    df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'SpsNum')

    df$value[df$SpsNum < 0] <- 0

    p <- ggplot() +
      geom_raster(data = df, aes(lon, lat, fill = SpsNum), interpolate = T) +
      geom_contour(data = df, aes(lon, lat, z = SpsNum)) +
      geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +
      scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(dat_tun$SpsNum))) + # all
      # maps same color scale

      geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
      color="grey40") +
      coord_equal(xlim = xlim, ylim = ylim) +
      ggtitle(paste0("SpsNum ",i, " TUNA")) +
      theme_bw() +
      theme(panel.background = element_rect(fill = "white"),
            panel.grid = element_blank(),
            plot.title = element_text(hjust=0.5),
            axis.title= element_blank(),
            legend.title=element_blank())
  }
}

```

```
png(filename  
paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  
RESULTS/MAPS/TUNA/TUN_SpsNum_548_",i,"_Feb20.png"),
```

```
width = 20, height = 15, units = "cm", res = 400)
```

```
print(p)
```

```
dev.off()
```

```
}
```

```
}
```

```
myfunction2(dat_tun)
```

```
# ----
```

```
# ---- TUNA (1-SIMPSON) INDEX SPATIAL DISTRIBUTION 2012-2016 ----
```

```
mba <- mba.surf(dat_tun[ , c('lon', 'lat', 'X1.Simps')], 200, 200)  
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)  
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'X1.Simps')  
df$value[df$X1.Simps < 0] <- 0
```

```
png(filename  
paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM  
RESULTS/MAPS/TUNA/TUN_1-Simpson_548_12-16.png"),
```

```
width = 20, height = 20, units = "cm", res = 400)
```

```
p <- ggplot() +
```

```
  geom_raster(data = df, aes(lon, lat, fill = X1.Simps), interpolate = T) +  
  geom_contour(data = df, aes(lon, lat, z = X1.Simps)) +  
  geom_point(data = data, aes(lon, lat, colour=year), shape=1, size = 3) +  
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(dat_tun$X1.Simps))) +
```

```
geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
color="grey40") +
```

```
coord_equal(xlim = xlim, ylim = ylim) +
```

```
ggtitle(paste0("1-Simpson 2012-2016 TUNA")) +
```

```
theme_bw() +
```

```
theme(panel.background = element_rect(fill = "white"),
```

```
panel.grid = element_blank(),
```

```
plot.title = element_text(hjust=0.5),
```

```
axis.title= element_blank(),
```

```
legend.title=element_blank())
```

```
print(p)
```

```
dev.off()
```

```
# ---- TUNA (1-SIMPSON) INDEX SPATIAL DISTRIBUTION BY YEAR ----
```

```
myfunction2 <- function(x){
```

```
for(i in 2012:2016){
```

```
temp <- x %>%
```

```
filter(year == i)
```

```
# mba.surf returns a surface approximated from a bivariate scatter of data points using
multilevel B-splines
```

```
mba <- mba.surf(temp[ , c('lon', 'lat', 'X1.Simps')], 200, 200)
```

```
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)
```

```
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'X1.Simps')
```

```
df$value[df$X1.Simps < 0] <- 0
```

```
p <- ggplot() +
```

```
geom_raster(data = df, aes(lon, lat, fill = X1.Simps), interpolate = T) +
```

```

geom_contour(data = df, aes(lon, lat, z = X1.Simps)) +
  geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(0, max(dat_tun$X1.Simps))) +
  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
  color="grey40") +
  coord_equal(xlim = xlim, ylim = ylim) +
  ggtitle(paste0("1-Simpson ", i, " TUNA")) +
  theme_bw() +
  theme(panel.background = element_rect(fill = "white"),
  panel.grid = element_blank(),
  plot.title = element_text(hjust=0.5),
  axis.title= element_blank(),
  legend.title=element_blank())

  png(filename =
  paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM
RESULTS/MAPS/TUNA/TUN_1-Simpson_548_", i, "_Feb20.png"),
  width = 20, height = 15, units = "cm", res = 400)
  print(p)
  dev.off()
}

}

myfunction2(dat_tun)
# ----

# ---- TUNA (1/SIMPSON) INDEX SPATIAL DISTRIBUTION 2012-2016 ----

mba <- mba.surf(dat_tun[ , c('lon', 'lat', 'inv_Simps')], 200, 200)
dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)

```

```
df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'inv_Simps')
```

```
png(filename = "TUNA_inv_Simpson_548_12-16.png",
     width = 20, height = 20, units = "cm", res = 400)
```

```
p <- ggplot() +
  geom_raster(data = df, aes(lon, lat, fill = inv_Simps), interpolate = T) +
  geom_contour(data = df, aes(lon, lat, z = inv_Simps)) +
  geom_point(data = data, aes(lon, lat, colour = year), shape=1, size = 3) +
  scale_fill_gradientn(colours = matlab.like2(7), limits = c(1, max(dat_tun$inv_Simps))) + # el
min de invSimps es siempre 1
  geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
color="grey40") +
  coord_equal(xlim = xlim, ylim = ylim) +
  ggtitle(paste0("1/Simpson 2012-2016 TUNA")) +
  theme_bw() +
  theme(panel.background = element_rect(fill = "white"),
        panel.grid = element_blank(),
        plot.title = element_text(hjust=0.5),
        axis.title= element_blank(),
        legend.title=element_blank())
print(p)
dev.off()
```

---- TUNA (1/SIMPSON) INDEX SPATIAL DISTRIBUTION BY YEAR ----

```

myfunction2 <- function(x){

  for(i in 2012:2016){

    temp <- x %>%
      filter(year == i)

    # mba.surf returns a surface approximated from a bivariate scatter of data points using
    # multilevel B-splines

    mba <- mba.surf(temp[ , c('lon', 'lat', 'inv_Simps')], 200, 200)

    dimnames(mba$xyz.est$z) <- list(mba$xyz.est$x, mba$xyz.est$y)

    df <- melt(mba$xyz.est$z, varnames = c('lon', 'lat'), value.name = 'inv_Simps')

    p <- ggplot() +
      geom_raster(data = df, aes(lon, lat, fill = inv_Simps), interpolate = T) +
      geom_contour(data = df, aes(lon, lat, z = inv_Simps)) +
      geom_point(data = temp, aes(lon, lat), colour = "black", size = .7) +
      scale_fill_gradientn(colours = matlab.like2(7), limits = c(1, max(dat_tun$inv_Simps))) +
      geom_polygon(data = spain, aes(x = long, y = lat, group = group), fill="light grey",
      color="grey40") +
      coord_equal(xlim = xlim, ylim = ylim) +
      ggtitle(paste0("1/Simpson ",i," TUNA")) +
      theme_bw() +
      theme(panel.background = element_rect(fill = "white"),
            panel.grid = element_blank(),
            plot.title = element_text(hjust=0.5),
            axis.title= element_blank(),
            legend.title=element_blank())

    png(filename
        paste0("C://Users/esther.barber/Desktop/DIVERPEL/DIVERPEL_R/GAM/GAM
RESULTS/MAPS/TUNA/TUN_inv_Simpson_548_",i,"_Feb20.png"),
        width = 20, height = 15, units = "cm", res = 400)
  }
}
  
```

```
print(p)
```

```
dev.off()
```

```
}
```

```
}
```

```
myfunction2(dat_tun)
```

```
# ----
```

```
# _____
```

```
# ---- [[[[[[[[[[[[[[[ THE END ]]]]]]]]]]]]]]] ----
```