

# **Marine Sensitivity**

## **Project Documentation**

Ben Best

2025-02-06

# Table of contents

<b>Preface</b>	<b>5</b>
<b>1 Introduction</b>	<b>6</b>
<b>I Science</b>	<b>7</b>
<b>2 Science</b>	<b>8</b>
<b>3 Stressors</b>	<b>9</b>
3.1 Offshore Wind Energy . . . . .	9
3.2 Oil & Gas . . . . .	10
<b>4 Receptors</b>	<b>11</b>
4.1 Species . . . . .	11
4.1.1 Corals . . . . .	11
4.1.2 Invertebrates . . . . .	11
4.1.3 Fish . . . . .	11
4.1.4 Marine Mammals . . . . .	11
4.1.5 Seabirds . . . . .	11
4.1.6 Sea Turtles . . . . .	11
4.2 Habitats . . . . .	11
4.2.1 Coral Reefs . . . . .	11
4.2.2 Hydrothermal Vents . . . . .	11
4.2.3 Kelp Forests . . . . .	11
4.2.4 Mangrove Forests . . . . .	11
4.2.5 Seamounts . . . . .	11
4.3 Primary Productivity . . . . .	11
<b>5 Exposure</b>	<b>13</b>
<b>6 Scoring</b>	<b>14</b>
6.1 Visualization . . . . .	14
6.1.1 Treemap . . . . .	14
6.1.2 Flower Plot . . . . .	15

<b>II Software</b>	<b>18</b>
<b>7 Software</b>	<b>19</b>
7.0.1 Interactive Applications . . . . .	19
7.0.2 Overcoming Challenges with Large Spatial Data . . . . .	20
7.0.3 Github Repositories . . . . .	21
7.0.4 Software Components . . . . .	23
<b>8 Server</b>	<b>24</b>
8.1 Setup . . . . .	24
8.1.1 launch instance . . . . .	24
8.1.2 ssh to server . . . . .	25
8.1.3 install docker . . . . .	26
8.1.4 Backup /share with snapshots . . . . .	27
8.2 Docker compose . . . . .	28
8.3 DNS . . . . .	32
8.4 Caddyfile . . . . .	32
8.5 Services . . . . .	33
<b>9 Database</b>	<b>37</b>
9.1 Table and Column Naming Conventions . . . . .	37
9.2 Species Distribution Models . . . . .	37
<b>10 Workflows</b>	<b>38</b>
<b>11 APIs</b>	<b>39</b>
<b>12 Libraries</b>	<b>41</b>
<b>13 Apps</b>	<b>42</b>
<b>14 Docs</b>	<b>43</b>
<b>15 Summary</b>	<b>44</b>
<b>References</b>	<b>45</b>
<b>Appendices</b>	<b>46</b>
<b>Glossary</b>	<b>46</b>

<b>III Applications</b>	<b>47</b>
<b>Areas of Interest</b>	<b>48</b>
<b>Bird Hotspots</b>	<b>49</b>
<b>Regional Map</b>	<b>50</b>
<b>Vulnerability Mapper</b>	<b>51</b>
<b>Distributions, Vector</b>	<b>52</b>
<b>Distributions, Raster</b>	<b>53</b>
<b>Vulnerability Mapper</b>	<b>54</b>

# Preface

This is a Quarto book.

# 1 Introduction

This Marine Sensitivity (MS) project of [BOEM](#) seeks to assess the sensitivity of marine species to offshore energy development, whether oil & gas or wind. By combining the best available species distributions with known species sensitivities we can map out areas of the ocean that are most vulnerable to human activities. This information can be used to inform decisions about where to place energy infrastructure and/or implement mitigations to minimize impacts on the marine environment.

This is a process, not a product. Information is imperfect, especially given the large expanse of US waters. Distributions and abundance of species change, modified increasingly by climate change and human activities. Knowledge on species sensitivities continues to expand with more research. And finally the methods for both modeling and distributing all this information continue to improve. We aim to provide a transparent and reproducible process that can be regularly updated as new data and methods become available.

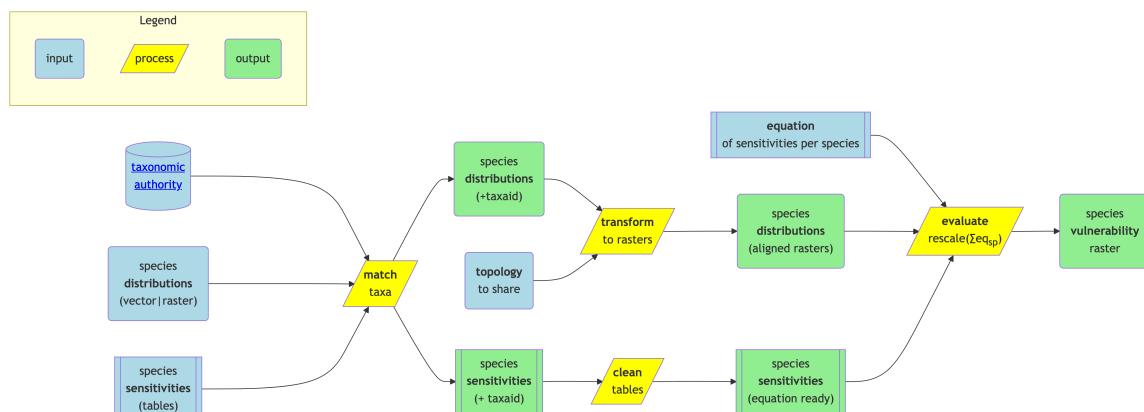


Figure 1.1: Flowchart of process for incorporating marine species sensitivities with distributions and generating a holistic vulnerability map.

# **Part I**

# **Science**

## 2 Science

The term vulnerability ( $V$ ) is a function of exposure ( $E$ ), sensitivity ( $S$ ) and adaptive capacity ( $A$ ) (Equation 2.1).

$$V = f(E, S, A) \quad (2.1)$$

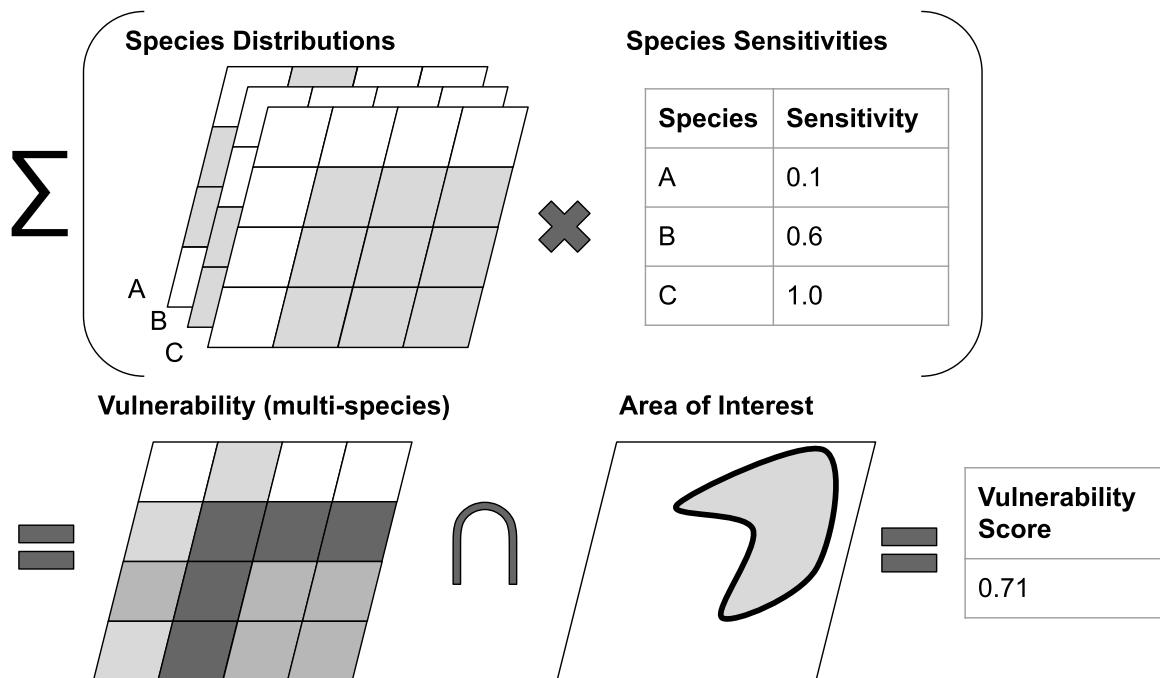


Figure 2.1: Overview of process.

$$cell_V = \sum_{spp} p * w \quad (2.2)$$

The raster of vulnerability ( $V$ ) contains cells representing a sum across species ( $spp$ ) of presence ( $p$ ) multiplied by the sensitivity weight ( $w$ ) (Equation 2.2).

# 3 Stressors

## 3.1 Offshore Wind Energy

Evaluation of stressors from the offshore wind industry needs to be evaluated based on human activities given the phase of development, whether pre-construction, construction, operation or decommissioning (Figure 3.1).

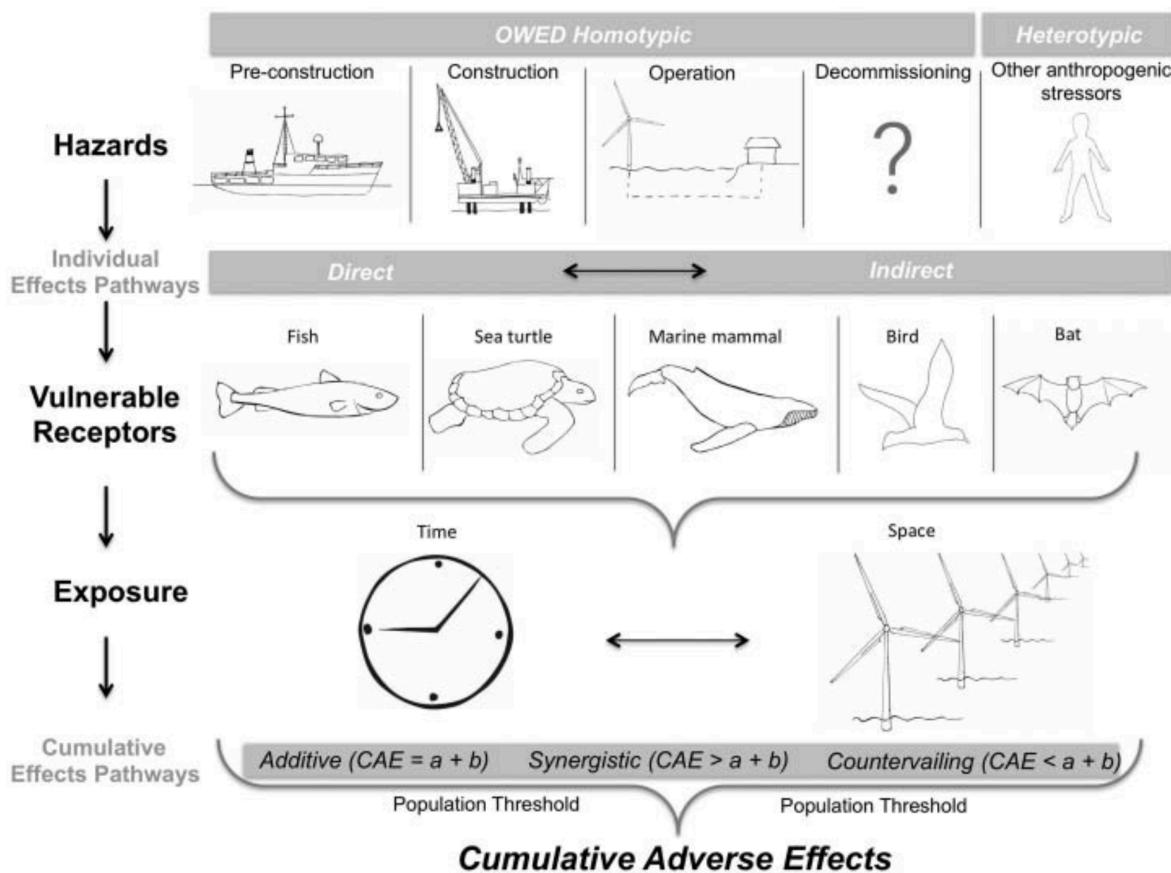


Figure 3.1: Cumulative adverse effects of offshore wind energy development on wildlife (Goodale and Milman 2016).

### **3.2 Oil & Gas**

# **4 Receptors**

Receptors are the species and habitats that are potentially impacted by the human activity.

## **4.1 Species**

### **4.1.1 Corals**

### **4.1.2 Invertebrates**

### **4.1.3 Fish**

### **4.1.4 Marine Mammals**

### **4.1.5 Seabirds**

### **4.1.6 Sea Turtles**

## **4.2 Habitats**

### **4.2.1 Coral Reefs**

### **4.2.2 Hydrothermal Vents**

### **4.2.3 Kelp Forests**

### **4.2.4 Mangrove Forests**

### **4.2.5 Seamounts**

## **4.3 Primary Productivity**

Primary productivity is specified in the explicit mandate for BOEM's management, per the Outer Continental Shelf Lands Act (OCSLA), Section 18(a)(2) of the OCSLA Amendments

of 1978 specifying 8 factors the USDOI must consider in the timing and location of OCS oil and gas activities, including “the relative environmental sensitivity and marine productivity of different areas of the OCS.” (Balcom et al. 2011)

We processed the Vertically Generalized Production Model (VGPM) product from [Oregon State’s Ocean Productivity Lab](#) (using the script `vg.R`) (Figure 4.1).

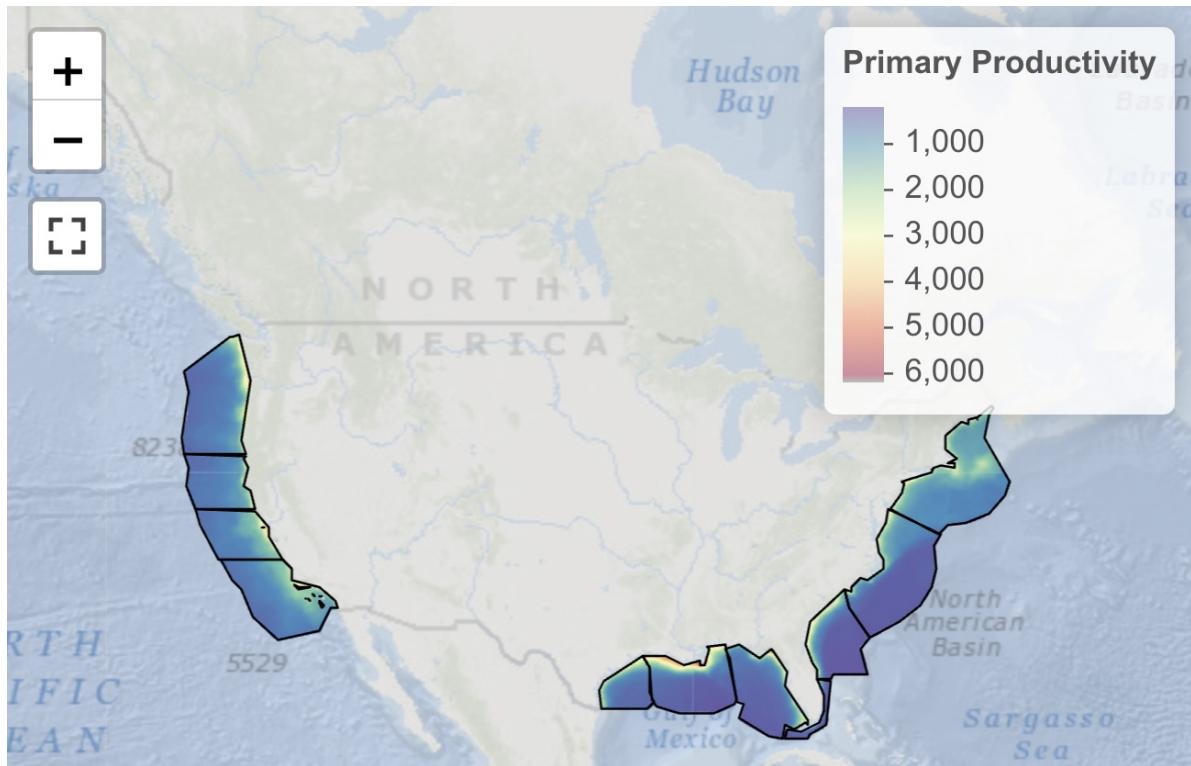


Figure 4.1: Primary productivity throughout the continental United States averaged across months of 2021, as measured by Oregon State’s Vertically Generalized Production Model (VGPM). Expansion is anticipated across the entire US EEZ and for more recent years.

## 5 Exposure

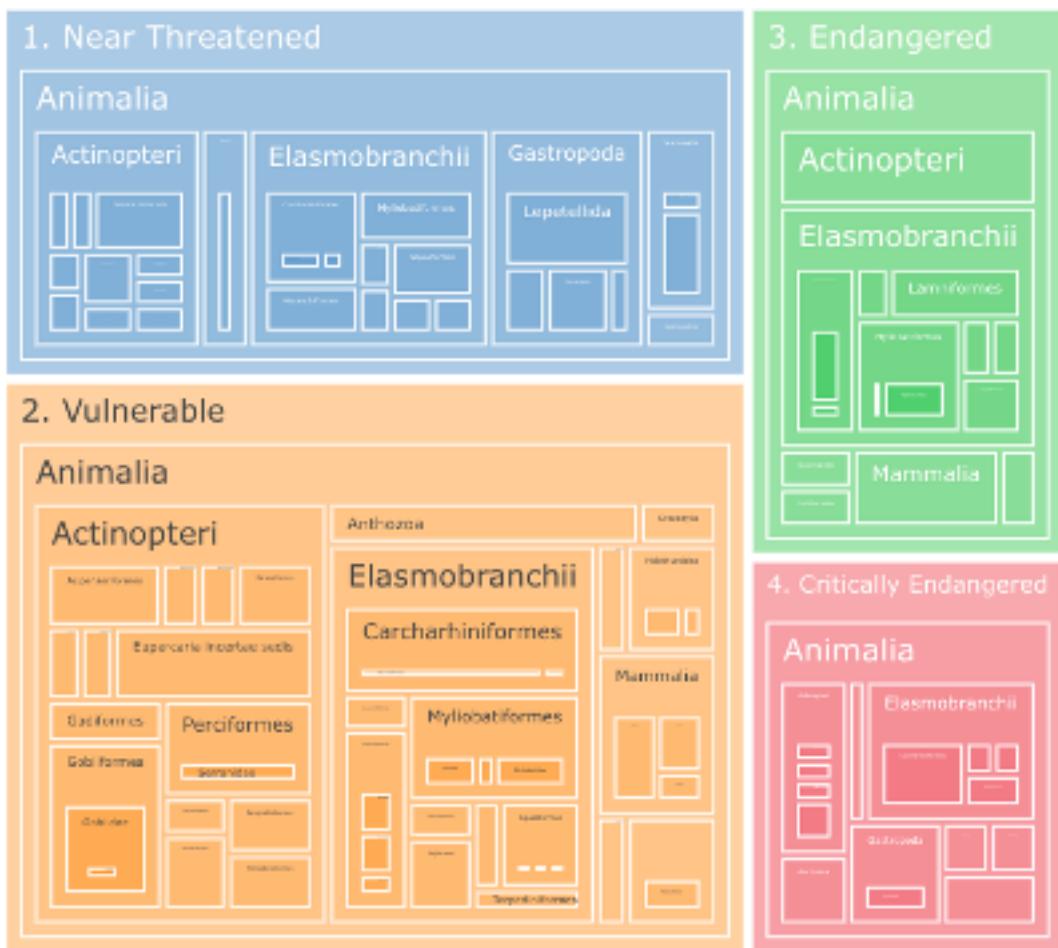
Cumulative exposure (Figure 3.1) is important for understanding impacts to a population.

# 6 Scoring

## 6.1 Visualization

We are in process with sorting the scoring methodology informed by visualization approaches.

### 6.1.1 Treemap



- Audience: scientists
- Show contribution of each element to a given pixel or area of interest
- Elements can be hierarchical across Sensitivity metrics and/or taxonomy
- Interactively zoom, e.g.: 4. Critically Endangered > Animalia > Mammalia



### 6.1.2 Flower Plot

- **Petal Length**

“One question was about the meaning of the numbers on the pedal plot. I explained that a higher sensitivity score indicates an area that could be susceptible to minor perturbations, while a lower score suggests an area that is more robust to minor changes.”  
- TW

- **OHI**

In the original Ocean Health Index (Halpern et al. 2012), the length of the petal reflected the percent towards maximum sustainability of the given goal. The framework is also based on a reference point, either spatially or temporally (Samhouri et al. 2012).

– RESA ’25

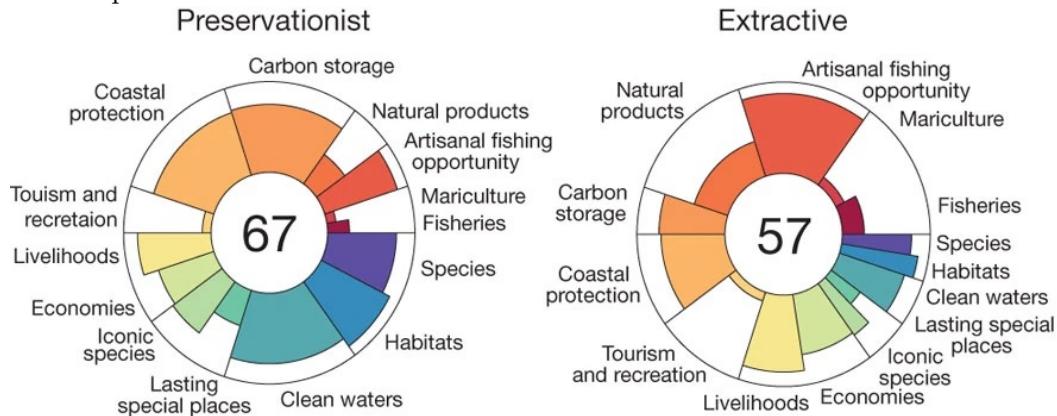
It makes sense to make the most vulnerable the highest score, so a low score is preferable (versus a preferred high score for sustainability of an OHI goal). We'll need to determine what the "highest" score means as a reference point and consider **SMART criteria**. (Specific, Measurable, Assignable, Realistic, Time-related). Will the reference point be the same globally or vary based on some regional maximum?

- Petal Width (and possibly varying weights)

“Another question was about how we manage the weights. I mentioned that we are not currently applying weights, but someone seemed concerned about it. We can discuss this further if needed.” - TW

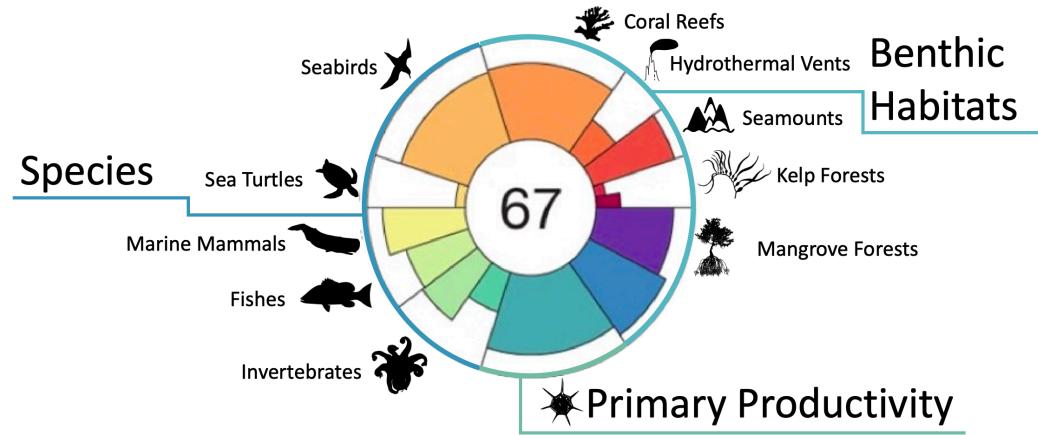
- OHI

For the OHI flower plot, the width of the petal represents its weight contributing to the weighted average score in the center of the flower. The varying importance of each goal is based on societally held values with potentially different value sets, such as preservationist vs extractive:



– RESA ’25

For the RESA '25, when looking at Species, Habitats and Primary Productivity, how shall they be weighted to factor into a final score? Shall species groups be simply weighted based on the number of species within, total biomass, extinction risk, commercial value, etc? And then Species versus less numerous Benthic Habitats and a singular value for Primary Productivity? I will conduct a literature survey to assign sensible value sets, but an App could also apply a user-defined value set to assign weights. That gets messy, but if transparent and easy ideally would lead to scientific consensus with a workshop and/or survey.



## **Part II**

# **Software**

# 7 Software

The **Marine Sensitivity Toolkit** (MST) is a stack of software components for reproducibly, interactively and hierarchically generating environmental vulnerability maps and scores. Tabular data is collated across studies evaluating sensitivity of species to oil & gas and offshore wind energy development. The best available species and benthic habitat distributions are being mosaicked across the US EEZ. Scores will be summarized according to subgroups within Species, Benthic Habitats and Primary Productivity. These scores will be averaged into an overall score and visualized as a flower plot, applicable to various levels of BOEM relevancy: regional, ecoregions, protraction diagrams, blocks and aliquots. The software components for achieving this interactively are: 1) server — a Docker configuration to spin up all the software; 2) database — a spatially enabled database (PostgreSQL with PostGIS extension); 3) workflows — scripts (as Quarto notebooks) to explore, ingest and update the database and output files; 4) APIs — application programming interfaces for rendering vector (using pg\_tileserv) or raster (using TiTiler) tile services for interactive mapping as well as a custom API (using R plumber); 5) libraries — re-usable documented functions as an R package for import, analysis and visualization using the APIs; 6) applications — interactive apps using the R Shiny framework for rendering vulnerability maps; and 7) documentation — as a Quarto notebook with figures, tables, glossary and references in interactive html or static docx/pdf output formats. This toolbox is intended to primarily serve BOEM needs internally, but by being open-source and fully reproducible the hope is to enlist buy-in and even contributions from external partners, whether from other government agencies, academia, NGOs or industry.

We ascribe to the philosophy of sharing all code for the sake of reproducibility, transparency and efficiency (Maitner et al. 2024; Lowndes et al. 2017); i.e. the FAIR principles of Findability, Accessibility, Interoperability, and Reusability (Wilkinson et al. 2016).

## 7.0.1 Interactive Applications

We have developed a series of interactive applications to explore the data and results of the MST project. These applications allow users to visualize the data, explore the results, and interact with the data in a more intuitive way. The applications are built using the `shiny` package in R (Chang et al. 2024), which allows us to easily create a user interface with complex reactivity for an interactive web application easily accessed through a web browser. The applications are designed to be user-friendly and intuitive, with interactive maps, charts, and tables that allow users to explore the data in a more dynamic way.

## 7.0.2 Overcoming Challenges with Large Spatial Data

The MS project incorporates many large spatial datasets that are problematic to render in a typical interactive application. For instance, the most common interactive mapping R package `leaflet` has a 4MB limitation for displaying rasters (see “Large Raster Warning” in [Raster Images • leaflet](#)). Vectors (i.e., points, lines and polygons) get smoothed when containing many vertices, but contiguity gets lost between polygons and rendering degrades to non-useable depending on the internet speed of the user’s connection.

To work around these limitations, we have implemented “cloud native” web services and formats (see also [Cloud-Optimized Geospatial Formats Guide](#)). Our implementations effectively reduce the size of any given spatial object based on the zoom level of the user’s browser. For rasters, we use cloud-optimized GeoTIFFs (COGs) and for vectors, we use Mapbox Vector Tiles (MVT). These formats are designed to be fast and efficient for web mapping applications, and they allow us to display large spatial datasets in an interactive web application without sacrificing performance or usability. Let’s take a closer look at implementation of each.

### 7.0.2.1 Raster: Cloud-Optimized GeoTIFFs (COGs) and Titiler

Historically, to read a raster, such as a GeoTIFF, from the web, the client software would have to read the entire file before rendering. Cloud Optimized GeoTIFFs ([COGs](#)) take advantage of [HTTP GET range requests](#) to read only the part of the file needed for rendering. So a COG stores quadtree simplifications of the original raster at multiple zoom levels and metadata for accessing their byte ranges in the file in the metadata header. This allows the client software to request only the parts of the file needed for rendering, which can greatly reduce the amount of data transferred and speed up rendering. This is for accessing the raw data in pixel values, e.g., for a raster of species distribution then the abundance of a species in each cell. We would want to also apply a color ramp to visualize the data. The open-source ([TiTiler](#)) software is a lightweight web service that serves up these color ramped tiles on the fly. So COGs can be stored on a simple file server (like Amazon S3 or Azure Blob Storage) and served up as interactive web maps with TiTiler as an intermediary between the COG files and the client accessing the interactive Shiny mapping app (Figure 7.1).

### 7.0.2.2 Vector: Mapbox Vector Tiles (MVTs) and pg\_tileserv

Although “cloud native” vector formats exist for simple file storage (see [Cloud-Optimized Geospatial Formats Guide](#)), none of these allow for flexible filtering and manipulation. Instead, we use PostgreSQL with the spatial extension ([PostGIS](#)) to store the vector data and serve it as Mapbox Vector Tiles ([MVTs](#)) using the [pg\\_tileserv](#) web service written in the language Go, which is very fast. This means that we don’t have to pre-render the MVTs (such as you might do with [tippecanoe](#)), but can instead serve the raw vector data directly from the database and let [pg\\_tileserv](#) handle the rendering on the fly. Filters (in the form of [CQL](#)) can be applied

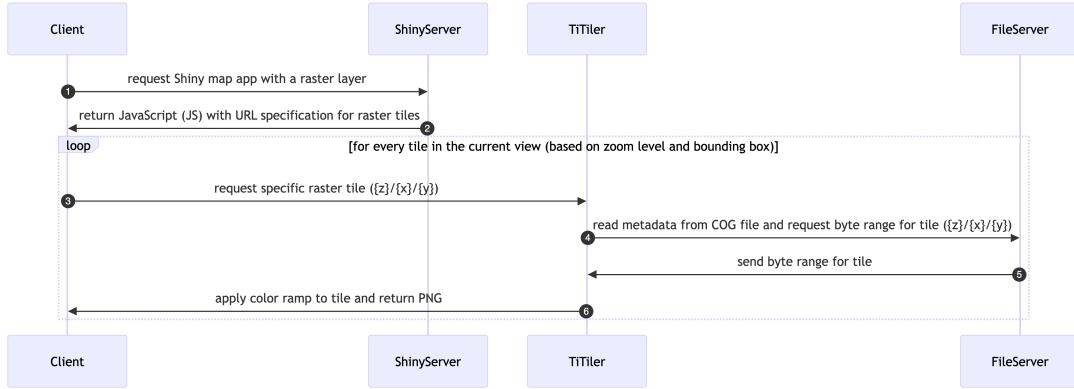


Figure 7.1: Sequence diagram implementing large raster interactive display using Cloud-Optimized GeotIFFs (COGs) and Titiler in a Shiny mapping app.

to the request. Symbology is rendered client-side via JavaScript, which allows for interactive hover and click events on vector objects (e.g., BOEM aliquot). Some speed-up is enabled by implementing a Varnish cache service in between. We can even write our own database functions for customized rendering, such as H3 hexagonal summaries. This allows us to serve vector data as web maps with minimal configuration and setup, and it provides a fast and efficient way to display large vector datasets in an interactive web application (Figure 7.2).

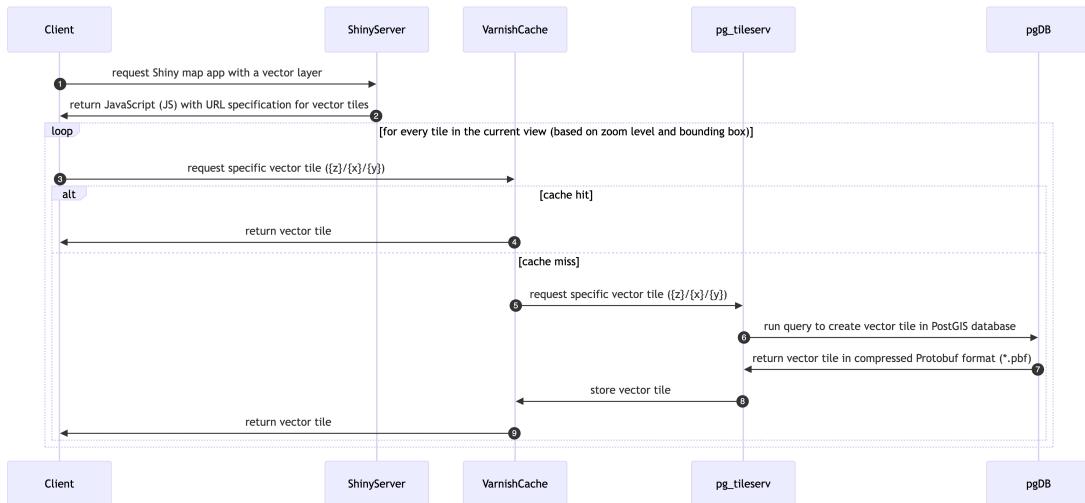
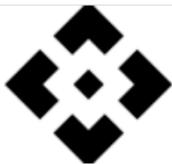


Figure 7.2: Sequence diagram implementing large vector interactive display using Mapbox Vector Tiles (MVTs) and pg\_tileserv in a Shiny mapping app.

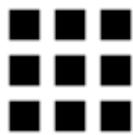
### 7.0.3 Github Repositories

repo	description
api	application programming interface (API) using R Plumber package
apps	Shiny applications
docs	documentation for BOEM's offshore environmental sensitivity index products
manuscripts	Manuscripts with review of sensitivities by industry and receptors (species, habitats, human uses)
MarineSensitivity.github.io	default website
msens	R library of functions for mapping marine sensitivities, sponsored by BOEM
objectives	repository for issues spanning multiple repositories and doing big picture roadmapping
server	server setup for R Shiny apps, RStudio IDE, R Plumber API, PostGIS database, pg_tileserv
workflows	scripts for testing data analytics and visualization as well as production workflows

#### 7.0.4 Software Components



APIs  
application programming  
interfaces (APIs)



Apps  
interactive applications using  
Shiny



Database  
PostgreSQL database  
extended spatially with  
PostGIS



Docs  
technical documentation



Libraries  
documented functions as an R  
package



Server  
server software configuration  
using Docker



Workflows  
scripts for exploring plus  
production workflows

# 8 Server

The server is for serving up any web services outside those of Github (e.g., [website](#), [docs](#) and R package [msens](#)) using [Docker](#) (see the [docker-compose.yml](#); with reverse proxying from subdomains to ports by [Caddy](#)).

## 8.1 Setup

For the latest instructions on launching an Amazon instance and installing the server software, see [Server Setup · MarineSensitivity/server Wiki](#), which is pasted below for convenience...

*on AWS as EC2 instance using Docker*

### 8.1.1 launch instance

name: **msens1**:

- Software Image (AMI)  
Canonical, **Ubuntu**, 22.04 LTS, amd64 jammy image build on 2023-09-19 ami-0fc5d935ebf8bc3bc
- Virtual server type (instance type)  
**t2.xlarge** (4 vCPU, 16 GB memory)
- Firewall (security group)  
New security group
- Storage (volumes)  
2 volume(s)
  - **20 GB**  
/ server software, disposable
  - **60 GB**  
`/share` for all data, persistent and to be backed up

### **8.1.1.1 allocate IP address**

- [Elastic IP addresses | EC2 | us-east-1](#) for persistent IP address
- Allocated IPv4 address: 100.25.173.0
- Associate Elastic IP address

### **8.1.2 ssh to server**

```
pem='/Users/bbest/My Drive/private/msens_key_pair.pem'  
ssh -i $pem ubuntu@msens1.marinesensitivity.org
```

#### **8.1.2.1 set hostname**

- Change the hostname of your Amazon Linux instance - [Amazon Elastic Compute Cloud](#)

```
sudo vi /etc/cloud/cloud.cfg  
# preserve_hostname: true  
sudo hostnamectl set-hostname msens1.marinesensitivity.org  
sudo reboot
```

#### **8.1.2.2 mount volume**

The extra volume (60 GB for `/share`) was added during EC2 launch instance wizard, but needs to be mounted before available for use.

- Make an Amazon EBS volume available for use on Linux - [Amazon Elastic Compute Cloud](#)

```
df -H
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/root	21G	2.3G	19G	11%	/
tmpfs	8.4G	0	8.4G	0%	/dev/shm
tmpfs	3.4G	898k	3.4G	1%	/run
tmpfs	5.3M	0	5.3M	0%	/run/lock
/dev/xvda15	110M	6.4M	104M	6%	/boot/efi
tmpfs	1.7G	4.1k	1.7G	1%	/run/user/1000

```
lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
loop0	7:0	0	24.6M	1	loop	/snap/amazon-ssm-agent/7528
loop1	7:1	0	55.7M	1	loop	/snap/core18/2790
loop2	7:2	0	63.5M	1	loop	/snap/core20/2015
loop3	7:3	0	111.9M	1	loop	/snap/lxd/24322
loop4	7:4	0	40.8M	1	loop	/snap/snapd/20092
xvda	202:0	0	20G	0	disk	
xvda1	202:1	0	19.9G	0	part	/
xvda14	202:14	0	4M	0	part	
xvda15	202:15	0	106M	0	part	/boot/efi
xvdb	202:16	0	60G	0	disk	

```
sudo file -s /dev/xvdb
# /dev/xvdb: data
```

So no file system on `/dev/xvdb` yet.

```
sudo mkfs -t xfs /dev/xvdb
sudo mkdir /share
sudo mount /dev/xvdb /share
```

```
sudo cp /etc/fstab /etc/fstab.orig
sudo blkid
# /dev/xvdb: UUID="bc766dfb-1c42-49cf-9320-2242a2d48a2e" BLOCK_SIZE="512" TYPE="xfs"
sudo vim /etc/fstab
# UUID=bc766dfb-1c42-49cf-9320-2242a2d48a2e /share xfs defaults,nofail 0 2

df -h
sudo umount /share ; df -h
sudo mount -a ; df -h
```

### 8.1.3 install docker

Following:

- Step-by-Step Guide to Install Docker on Ubuntu in AWS | by Srija Anaparthys | Medium

```
sudo apt-get update
#OLD: sudo apt-get install docker.io -y
```

NEW: [[Migrate to docker compose]]

```
sudo systemctl start docker
sudo docker run hello-world
sudo systemctl enable docker
docker --version
# Docker version 24.0.6, build ed223bc
sudo usermod -a -G docker $(whoami)
```

#### 8.1.3.1 run docker compose

- /Users/bbest/My Drive/private/[msens\\_server\\_env-password.txt](#)

```
sudo chown -R ubuntu:ubuntu /share
mkdir -p /share/github/MarineSensitivity
cd /share/github/MarineSensitivity
# clone server repo
git clone https://github.com/MarineSensitivity/server.git
cd server

# add password, used as $PASSWORD in docker-compose.yml
echo 'PASSWORD=*****' > .env

# launch docker instances
sudo docker-compose up -d
```

#### 8.1.4 Backup /share with snapshots

Per [Automate snapshot lifecycles - Amazon Elastic Compute Cloud](#), created two policies:

- [bkup\\_msens-share\\_daily](#) every 24 hrs at 09:00 UTC, max of 7
- [bkup\\_msens-share\\_weekly](#) every Monday 09:00 UTC, max of 8

## 8.2 Docker compose

The Docker compose file is used to define and run multi-container Docker applications. Here is the [docker-compose.yml](#) file for the server pasted for convenience ...

```
version: "3.9"

services:
  caddy:
    container_name: caddy
    image: caddy:latest
    ports:
      - 80:80
      - 443:443
    restart: unless-stopped
    volumes:
      - ./caddy/Caddyfile:/etc/caddy/Caddyfile
      - /share:/share
      - /share/caddy/data:/data
      - /share/caddy/config:/config

  rstudio:
    container_name: rstudio
    build: ./rstudio
    environment:
      ROOT: 'true'
      USER: admin
      PASSWORD: ${PASSWORD}
      ADD: shiny
    ports:
      - 8787:8787 # rstudio
      - 3838:3838 # shiny
    restart: unless-stopped
    volumes:
      - /share:/share
      - /share/shiny_apps:/srv/shiny-server

  plumber:
    container_name: plumber
    build: ./plumber
    ports:
      - 8888:8888 # api
```

```

restart: unless-stopped
volumes:
  - /share:/share
depends_on:
  - postgis

postgis:
  container_name: postgis
  image: postgis/postgis:latest
  environment:
    POSTGRES_DB: msens
    POSTGRES_USER: admin
    POSTGRES_PASSWORD: ${PASSWORD}
    ANON_PASSWORD: ${ANON_PASSWORD}
    PGDATA: /share/postgis/data
  volumes:
    # all files in /docker-entrypoint-initdb.dare automatically executed
    # in alphabetical order on container creation
    - ./postgis/init.sh:/docker-entrypoint-initdb.d/init.sh # add user anon
    - /share:/share
    - /share/postgis:/var/lib/postgresql
  restart: unless-stopped
  healthcheck:
    test: 'exit 0'
  ports:
    - 5432:5432

pgadmin:
  container_name: pgadmin
  image: dpage/pgadmin4:8.14
  restart: always
  environment:
    PGADMIN_DEFAULT_EMAIL: ben@ecoquants.com
    PGADMIN_DEFAULT_PASSWORD: ${PASSWORD}
    PGADMIN_LISTEN_PORT: 8088
    # PGADMIN_CONFIG_CONFIG_DATABASE_URI: "'postgresql://admin:$PASSWORD@postgis:5432/msens"
  ports:
    - 8088:8088
  volumes:
    - /share/pgadmin:/var/lib/pgadmin
depends_on:
  - postgis

```

```

pgbkups:
  container_name: pgbkups
  image: prodigestivill/postgres-backup-local
  restart: always
  user: postgres:postgres # Optional: see below
  volumes:
    - /share/postgis_backups:/backups
      # sudo mkdir /share/postgis_backups; sudo chown -R 999:999 /share/postgis_backups
  links:
    - postgis
  depends_on:
    - postgis
  environment:
    - POSTGRES_HOST=postgis
    - POSTGRES_DB=msens
    - POSTGRES_USER=admin
    - POSTGRES_PASSWORD=${PASSWORD}
    - POSTGRES_EXTRA_OPTS=-Z6 --blobs
    - SCHEDULE=@daily
    - BACKUP_KEEP_DAYS=7
    - BACKUP_KEEP_WEEKS=4
    - BACKUP_KEEP_MONTHS=6
    - HEALTHCHECK_PORT=8088

tile:
  container_name: tile
  environment:
    DATABASE_URL: 'postgresql://admin:${PASSWORD}@postgis:5432/msens'
  image: pramsey/pg_tileserv:latest
  depends_on:
    - postgis
  ports:
    - 7800:7800

tilecache:
  container_name: tilecache
  image: varnish:latest # 7.4.2 # last updated: 2023-12-26
  volumes:
    - /share:/share
      # - "./varnish/default.vcl:/etc/varnish/default.vcl"
  ports:
    - 6081:6081

```

```

environment:
  # VARNISH_SIZE: '2G'
  VARNISH_BACKEND_HOST: tile # .marinesensitivity.org
  VARNISH_BACKEND_PORT: 7800
  VARNISH_HTTP_PORT: 6081 # VARNISH_PROXY_PORT: '6081'
#command: "-p default_keep=43200" # 60*60*12 = 43200 sec = 12 hrs
restart: always
depends_on:
  - "tile"

rest:
  container_name: rest
  environment:
    PGRST_DB_URI: 'postgresql://anon:${ANON_PASSWORD}@postgis:5432/msens'
    PGRST_OPENAPI_SERVER_PROXY_URI: http://127.0.0.1:3000
    PGRST_DB_ANON_ROLE: anon # db-anon-role
  image: postgrest/postgrest
  depends_on:
    - postgis
  ports:
    - "3000:3000"

swagger:
  container_name: swagger
  image: swaggerapi/swagger-ui
  depends_on:
    - rest
  ports:
    - "8080:8080"
  expose:
    - "8080"
  environment:
    API_URL: https://rest.MarineSensitivity.org/

titiler:
  container_name: titiler
  image: ghcr.io/developmentseed/titiler:latest
  environment:
    PORT: 8000
    # WORKERS_PER_CORE: 1
  ports:
    - "8000:8000"

```

## 8.3 DNS

The domain name server (DNS) records are managed by [SquareSpace](#). The subdomains point to the server on Amazon at 100.25.173.0, whereas the main website is hosted by Github servers, per [Managing a custom domain for your GitHub Pages site - GitHub Docs](#).

Host	Type	Data
@	A	185.199.111.153
@	A	185.199.110.153
@	A	185.199.109.153
@	A	185.199.108.153
api	A	100.25.173.0
file	A	100.25.173.0
msens1	A	100.25.173.0
pgadmin	A	100.25.173.0
rest	A	100.25.173.0
rstudio	A	100.25.173.0
shiny	A	100.25.173.0
swagger	A	100.25.173.0
tile	A	100.25.173.0
tilecache	A	100.25.173.0
titiler	A	100.25.173.0
www	CNAME	marinesensitivity.org

## 8.4 Caddyfile

The Caddyfile parameterizes the reverse proxying between the external subdomains and the Docker's internal ports. Here is the [Caddyfile](#) pasted for convenience ...

```
api.marinesensitivity.org {
    reverse_proxy plumber:8888
}

file.marinesensitivity.org {
    root * /share/public
    file_server browse {
        # serve *.zst, *.br or *.gz if file exists and client supports precompressed files
}
```

```
    precompressed zstd br gzip
  }
}

pgadmin.marinesensitivity.org {
  reverse_proxy pgadmin:8088
}

rest.marinesensitivity.org {
  reverse_proxy rest:3000
}

rstudio.marinesensitivity.org {
  reverse_proxy rstudio:8787
}

shiny.marinesensitivity.org {
  reverse_proxy rstudio:3838
}

swagger.marinesensitivity.org {
  reverse_proxy swagger:8080
}

tile.marinesensitivity.org {
  reverse_proxy tile:7800
}

tilecache.marinesensitivity.org {
  reverse_proxy tilecache:6081
}

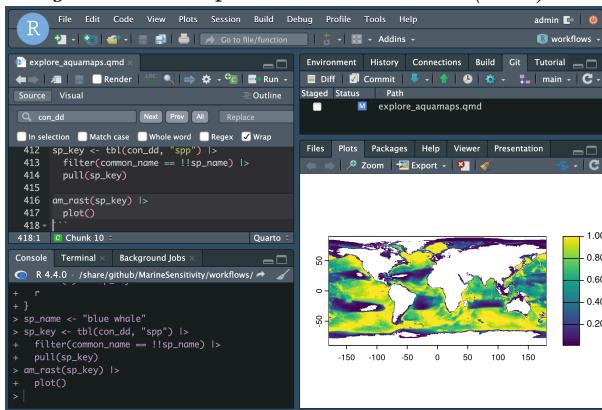
titiler.marinesensitivity.org {
  reverse_proxy titiler:8000
}
```

## 8.5 Services

The server is running the following services:

- **RStudio**

*integrated development environment (IDE) to code and debug directly on the server*

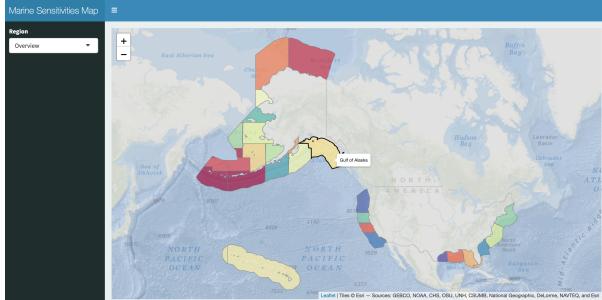


[More info..](#)

- **Shiny**

*interactive applications*

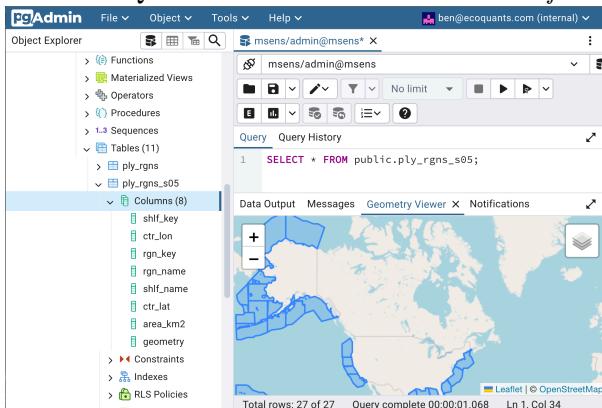
e.g., [shiny.marinesensitivity.org/map](http://shiny.marinesensitivity.org/map)



[More info..](#)

- **PGadmin**

*PostGreSQL database administration interface*



[More info..](#)

- **api**

*custom API: using R plumber*

The screenshot shows the MarineSensitivities Custom API documentation generated by Swagger. At the top, it says "MarineSensitivities Custom API 1.0.0 OAS3". Below that is a "API Description" section. Under "Servers", the URL "https://api.marinesensitivities.org/" is selected. The main content area shows a "default" endpoint with the following methods:

- GET /echo**: Echo back the input
- GET /plot**: Plot a histogram
- POST /sum**: Return the sum of two numbers
- GET /**: redirect to the swagger interface

[More info..](#)

- **swagger**

*generic database API: using PostGREST*

The screenshot shows the standard public schema PostGREST Documentation generated by Swagger. At the top, it says "standard public schema 1.0.0 OAS 2.0". Below that is a "PostGREST Documentation" section. Under "Schemes", "HTTP" is selected. The main content area shows an "Introspection" endpoint with the following methods:

- GET /**: OpenAPI description (the document)
- geography\_columns** (under "geography\_columns" section):
  - GET /geography\_columns**
- geometry\_columns** (under "geometry\_columns" section):
  - GET /geometry\_columns**
  - POST /geometry\_columns**
  - DELETE /geometry\_columns**
  - PATCH /geometry\_columns**

[More info..](#)

- **tile**

*spatial database API: using pg\_tileserv for serving vector tiles*

## pg\_tileserv

### Service Metadata

- [index.json](#) for layer list

### Table Layers

- aquamaps.cells ([preview](#) | [json](#))
- public.ply\_rgn ([preview](#) | [json](#))
- public.ply\_rgn\_s05 ([preview](#) | [json](#))
- public.ply\_shfs ([preview](#) | [json](#))
- public.ply\_shfs\_s05 ([preview](#) | [json](#))
- public.sdm\_geometries ([preview](#) | [json](#))
- raw.boem\_ak\_blk\_clip ([preview](#) | [json](#))
- raw.boem\_ak\_prot\_clip ([preview](#) | [json](#))
- raw.boem\_atl\_als ([preview](#) | [json](#))
- raw.boem\_atl\_blk\_clip ([preview](#) | [json](#))
- raw.boem\_atl\_prot\_clip ([preview](#) | [json](#))
- raw.boem\_gom\_blk\_clip ([preview](#) | [json](#))
- raw.boem\_gom\_prot\_clip ([preview](#) | [json](#))
- raw.boem\_pc\_als ([preview](#) | [json](#))
- raw.boem\_pc\_blk\_clip ([preview](#) | [json](#))
- raw.boem\_pc\_prot\_clip ([preview](#) | [json](#))
- raw.boem\_usa\_mhk\_plan ([preview](#) | [json](#))
- raw.boem\_usa\_wind\_lease ([preview](#) | [json](#))
- raw.boem\_usa\_wind\_plan ([preview](#) | [json](#))
- raw.mr\_eez ([preview](#) | [json](#))

### Function Layers

- [public.sdm\\_spatial](#) ([preview](#) | [json](#))

Serves the Species Distribution Model given parameters: dataset\_key, species\_key, popn, time\_interval, variable.

[More info..](#)

•••

# 9 Database

## 9.1 Table and Column Naming Conventions

- Table names are plural and use all lower case.
- Unique identifiers are suffixed with:
  - \*\_id for unique integer keys;
  - \*\_key for unique string keys;
  - \*\_seq for auto-incrementing sequence integer keys.
- Column names are singular and use snake\_case.
- Foreign keys are named with the singular form of the table they reference, followed by \_id.
- Primary keys are named id.

## 9.2 Species Distribution Models

See entity relationship diagram (ERD) for the species distribution models (SDM) database tables in this workflow:

- [Create SDM Tables](#)

And example of ingesting SDM outputs into the database in this workflow:

- [Ingest GoMex cetacean & sea turtle SDMs](#)

# 10 Workflows

Workflows are scripts for testing data analytics and visualization as well as production workflows for ingesting data. See:

- [marinesensitivity.org/workflows](https://marinesensitivity.org/workflows)  
rendered html pages from the scripts (as Quarto notebooks)
- [github.com/MarineSensitivity/workflows](https://github.com/MarineSensitivity/workflows)  
source code in the Github repository

# 11 APIs

There three APIs, each used for different purposes:

## 1. api

*custom API: using R plumber*  
source: [MarineSensitivity/api](#)

The screenshot shows the MarineSensitivity Custom API documentation generated by Swagger. At the top, there's a navigation bar with the title "MarineSensitivity Custom API 1.0.0 OAS3", the URL "https://api.marinesensitivities.org/openapi.json", and an "Explore" button. Below the title, there's a "Servers" dropdown set to "https://api.marinesensitivities.org/". The main content area is titled "default" and contains four API endpoints:

- GET /echo**: Echo back the input.
- GET /plot**: Plot a histogram.
- POST /sum**: Return the sum of two numbers.
- GET /**: redirect to the swagger interface.

## 2. swagger

*generic database API: using PostGREST*  
source: Postgres database, non-spatial

The screenshot shows the standard public schema PostGREST documentation generated by Swagger. At the top, there's a navigation bar with the title "standard public schema 1.0.0 OAS3", the URL "https://rest.MarineSensitivity.org/", and an "Explore" button. Below the title, there's a "Schemes" dropdown set to "HTTP". The main content area is titled "Introspection" and contains one endpoint:

- GET /**: OpenAPI description (No document)

Below "Introspection", there are sections for "geography\_columns" and "geometry\_columns", each with their own set of endpoints:

- geography\_columns**:
  - GET /geography\_columns**
- geometry\_columns**:
  - GET /geometry\_columns**
  - POST /geometry\_columns**
  - DELETE /geometry\_columns**
  - PATCH /geometry\_columns**

## 3. tile

*spatial database API: using pg\_tileserv for serving vector tiles*

source: Postgres database, spatial

## pg\_tileserv

### Service Metadata

- [index.json](#) for layer list

### Table Layers

- aquamaps.cells ([preview](#) | [json](#))
- public.ply\_rgns ([preview](#) | [json](#))
- public.ply\_rgns\_s05 ([preview](#) | [json](#))
- public.ply\_shfts ([preview](#) | [json](#))
- public.ply\_shfts\_s05 ([preview](#) | [json](#))
- public.sdm\_geometries ([preview](#) | [json](#))
- raw.boem\_ak\_blk\_clip ([preview](#) | [json](#))
- raw.boem\_ak\_prot\_clip ([preview](#) | [json](#))
- raw.boem\_atl\_atlq ([preview](#) | [json](#))
- raw.boem\_atl\_blk\_clip ([preview](#) | [json](#))
- raw.boem\_atl\_prot\_clip ([preview](#) | [json](#))
- raw.boem\_gom\_blk\_clip ([preview](#) | [json](#))
- raw.boem\_gom\_prot\_clip ([preview](#) | [json](#))
- raw.boem\_pc\_atq ([preview](#) | [json](#))
- raw.boem\_pc\_blk\_clip ([preview](#) | [json](#))
- raw.boem\_pc\_prot\_clip ([preview](#) | [json](#))
- raw.boem\_usa\_mnh\_plan ([preview](#) | [json](#))
- raw.boem\_usa\_wind\_lease ([preview](#) | [json](#))
- raw.boem\_usa\_wind\_plan ([preview](#) | [json](#))
- raw.mr\_eez ([preview](#) | [json](#))

### Function Layers

- [public.sdm\\_spatial](#) ([preview](#) | [json](#))

Serves the Species Distribution Model given parameters: dataset\_key, species\_key, popn, time\_interval, variable.

## 12 Libraries

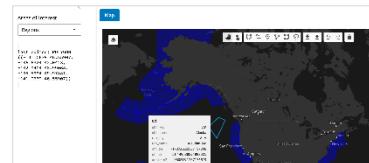
By creating an R package, we can document functions and make them easily available to other users.

- [msens](#)

R library of functions for mapping marine sensitivities, sponsored by BOEM

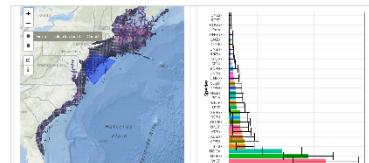
Functions can *read* data from the one [APIs](#) (which communicate with the [Database](#)), *analyze* the data, *visualize* the results and store some smaller *data*.

# 13 Apps



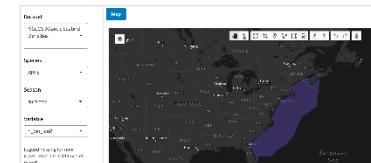
**Areas of Interest**  
AREAS VECTOR

Map high resolution Areas of Interest (using vector tiles) for visualization (and later summarization).



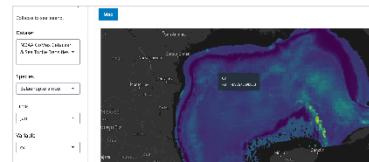
**Bird Hotspots**  
DISTRIBUTIONS HOTSPOTS

Bird hotspots application showing hotspot probability for species present given drawn Area of Interest.



**Distributions, Raster**  
DISTRIBUTIONS RASTER

Show species distributions with high resolution rasters as cloud-optimized GeoTIFFs (COGs).



**Distributions, Vector**  
DISTRIBUTIONS VECTOR

Show species distributions with high resolution vectors (as vector tiles).



**Regional Map**  
AREAS

Basic interactive map of BOEM regions.



**Vulnerability Mapper**  
DISTRIBUTIONS VULNERABILITIES  
RASTER

Combine species distribution models (raster) and vulnerability metrics (tables) to identify areas of high conservation concern.

See also details of individual applications in the Appendix.

## 14 Docs

Technical documentation is principally in this book:

- [marinesensitivity.org/docs](http://marinesensitivity.org/docs)  
the main documentation site
- [github.com/MarineSensitivity/docs](https://github.com/MarineSensitivity/docs)  
source code in the Github repository

But there are also some other self-documenting resources:

- [marinesensitivity.org/msens](http://marinesensitivity.org/msens)  
documented R functions

## **15 Summary**

...

# References

- Balcom, Brian J., Douglas C. Biggs, Chuanmin Hu, Paul Montagna, and Dean A. Stockwell. 2011. "A Comparison of Marine Productivity Among Outer Continental Shelf Planning Areas. Final Report."
- Chang, Winston, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. 2024. "Shiny: Web Application Framework for r." <https://CRAN.R-project.org/package=shiny>.
- Goodale, M. Wing, and Anita Milman. 2016. "Cumulative Adverse Effects of Offshore Wind Energy Development on Wildlife." *Journal of Environmental Planning and Management* 0 (0): 1–21. <https://doi.org/10.1080/09640568.2014.973483>.
- Halpern, Benjamin S., Catherine Longo, Darren Hardy, Karen L. McLeod, Jameal F. Samhouri, Steven K. Katona, Kristin Kleisner, et al. 2012. "An Index to Assess the Health and Benefits of the Global Ocean." *Nature*. <https://doi.org/10.1038/nature11397>.
- Lowndes, Julia S. Stewart, Benjamin D. Best, Courtney Scarborough, Jamie C. Afflerbach, Melanie R. Frazier, Casey C. O'Hara, Ning Jiang, and Benjamin S. Halpern. 2017. "Our Path to Better Science in Less Time Using Open Data Science Tools." *Nature Ecology & Evolution* 1 (6): 0160. <https://doi.org/10.1038/s41559-017-0160>.
- Maitner, Brian, Paul Efren Santos Andrade, Luna Lei, Jamie Kass, Hannah L. Owens, George C. G. Barbosa, Brad Boyle, et al. 2024. "Code Sharing in Ecology and Evolution Increases Citation Rates but Remains Uncommon." *Ecology and Evolution* 14 (8): e70030. <https://doi.org/10.1002/ece3.70030>.
- Ross, Pauline M., Elliot Scanes, Maria Byrne, Tracy D. Ainsworth, Jennifer M. Donelson, Shawna A. Foo, Pat Hutchings, Vengatesen Thiagarajan, and Laura M. Parker. 2023. "Surviving the Anthropocene: The Resilience of Marine Animals to Climate Change." In. CRC Press.
- Samhouri, Jameal F., Sarah E. Lester, Elizabeth R. Selig, Benjamin S. Halpern, Michael J. Fogarty, Catherine Longo, and Karen L. McLeod. 2012. "Sea Sick? Setting Targets to Assess Ocean Health and Ecosystem Services." *Ecosphere* 3 (5): 1–18. <https://doi.org/10.1890/es11-00366.1>.
- Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data* 3 (1): 160018. <https://doi.org/10.1038/sdata.2016.18>.

# Glossary

**acclimatisation** the adjustment of an organism to environmental conditions in the field or environment rather than the laboratory without an adjustment in their genetics. Acclimation has been used to describe phenotypically plastic responses in natural conditions. Source: Ross et al. (2023).

**adaptation** the evolutionary mechanism where natural selection of traits is genetically passed on, typically over many generations, to create an organism suited to the environment. Source: [rossRoss et al. (2023)

**adaptive capacity** the capacity of the ecosystem or organism to improve and reorganise in response to stress such as climate change through phenotypic plasticity (acclimation, acclimatisation) or adaptation, distributional shifts, and rapid evolution of traits suited to new conditions. Source: Ross et al. (2023).

**epigenetics** the modification of phenotype plasticity of an organism through altered gene expression without an alteration to the DNA sequence. ‘Epi’ means above the DNA and includes DNA methylation, modification of histones, and non-coding RNA. Source: Ross et al. (2023).

**exposure** the magnitude of the change in the environment

**fecundity** the maximum physiological potential reproductive output of an organism to produce offspring (reproductive output). This differs from fertility, which is the number of offspring born. Source: Ross et al. (2023)

**MBON** Marine Biodiversity Observation Network; see [MarineBON.org](#)

**resilience** the capacity of an ecosystem, society, or organism to absorb disturbance and reorganise while undergoing change so as to retain essentially the same function, structure, identity, and feedbacks. Resilience reflects the degree to which a complex adaptive system is determined by its capacity to reorganise and adapt in order to avoid being disturbed again. Source: Ross et al. (2023).

**sensitivity** the magnitude of response to the change

**stressor** the stimulus that causes stress to an organism

**vulnerability** combination of exposure and sensitivity

# **Part III**

# **Applications**

# Areas of Interest

Map high resolution Areas of Interest (using vector tiles) for visualization (and later summarization).

- [website](#)
- [code](#)

Area Explorer

The screenshot shows the 'Area Explorer' interface. On the left, there's a sidebar titled 'Areas of Interest' with a dropdown menu set to 'Regions'. Below it, a code snippet shows the coordinates of a polygon:

```
last edited: POLYGON((-141.0639 46.55907, -136.6324 42.0423, -132.0404 48.99068, -135.5994 49.97881, -141.0639 46.55907))
```

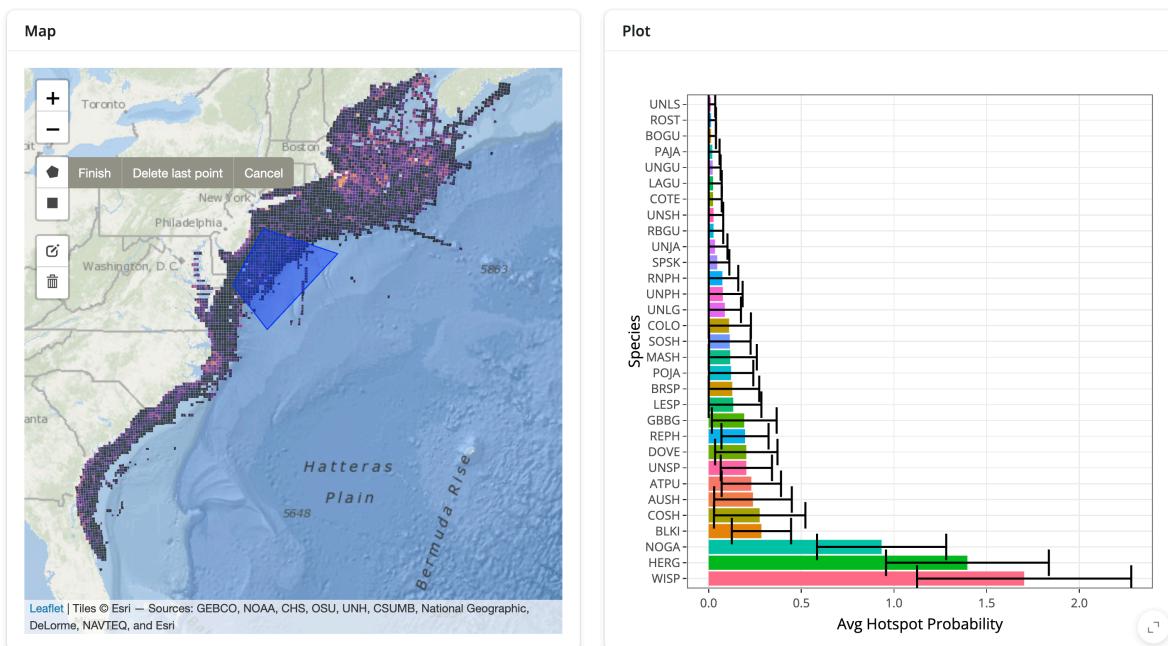
The main area is a map of North America with a blue polygon highlighting the Aleutian Arc region off the coast of Alaska. A tooltip provides detailed information about this area:

aoi	shlf_key	shlf_name	rgn_key	rgn_name	AK	Alaska	ALA
				Aleutian Arc			
	ctr_lon	-178.56252307137368					
	ctr_lat	51.149538291898025					
	area_km2	860518.1237176519					
	geometry						null

# Bird Hotspots

Bird hotspots application showing hotspot probability for species present given drawn Area of Interest.

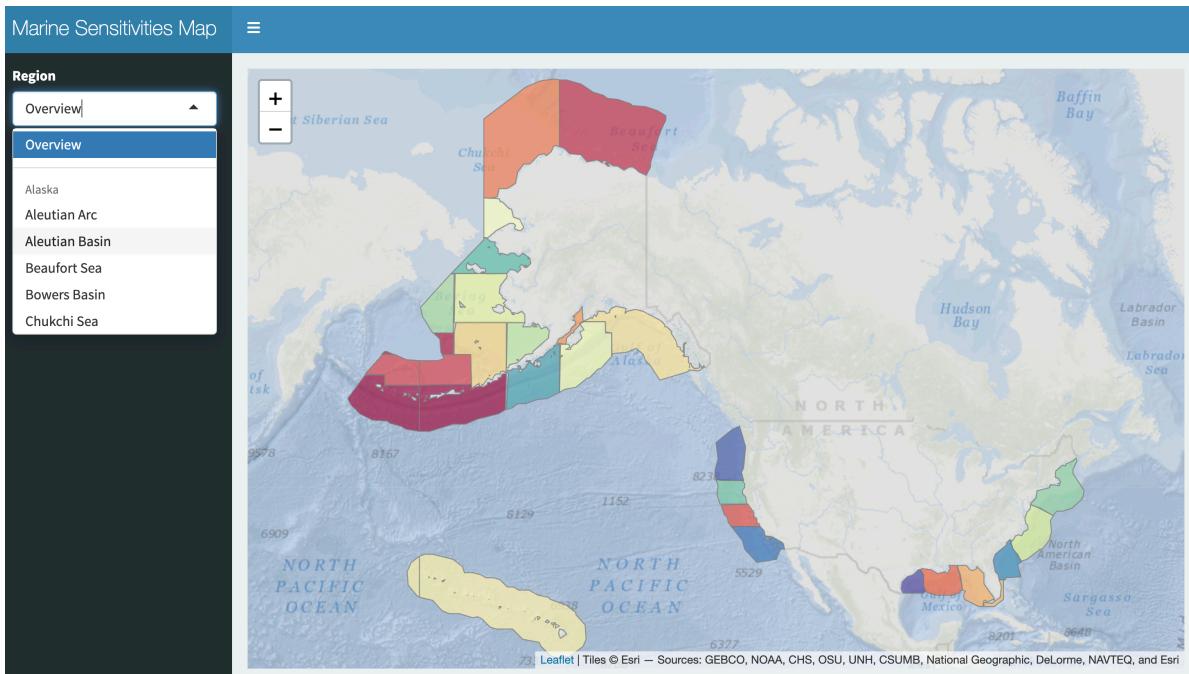
- [🌐 website](#)
- [💻 code](#)



# Regional Map

Basic interactive map of BOEM regions.

- [🌐 website](#)
- [🔗 code](#)

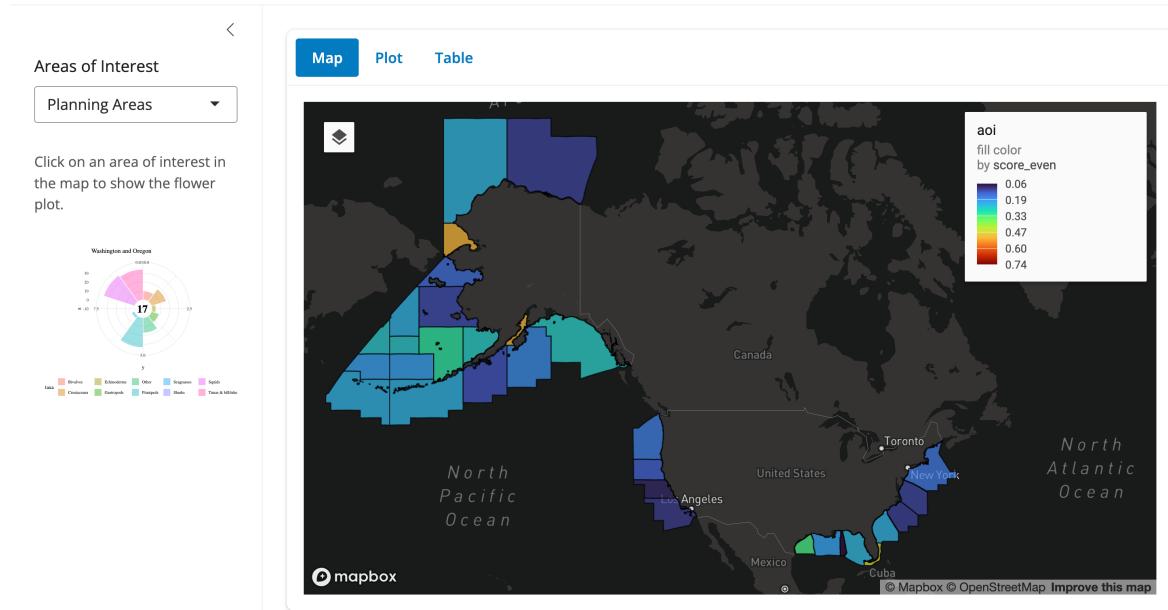


# Vulnerability Mapper

Calculate scores based on sensitivity metric – for now, extinction risk across taxonomic groups.

-  website
-  code

Scores Explorer

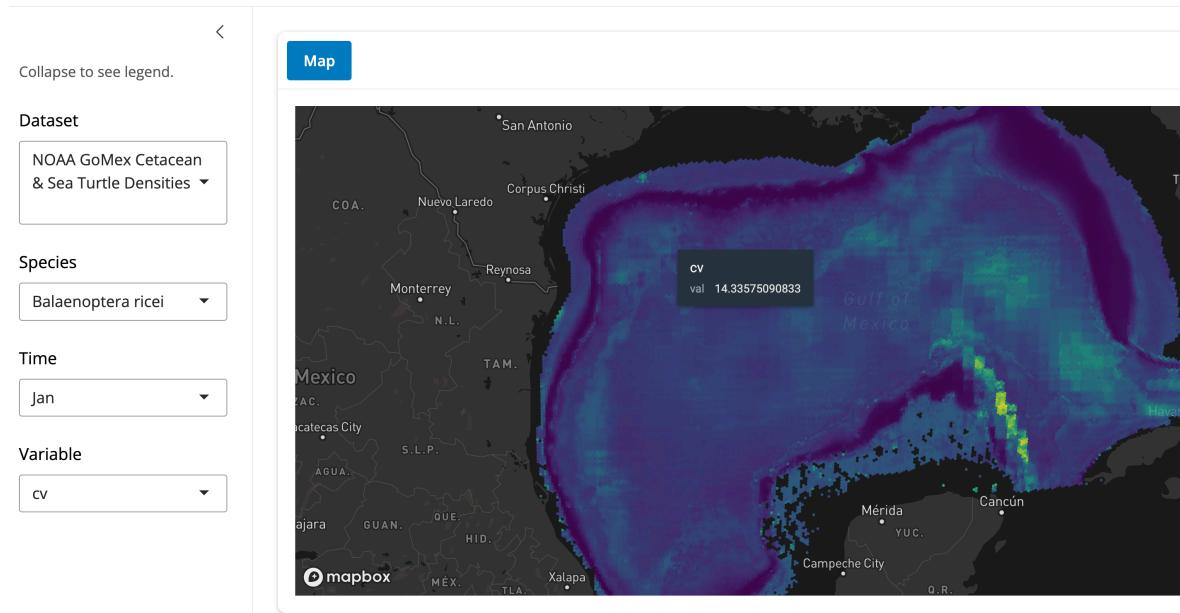


# Distributions, Vector

Show species distributions with high resolution vectors (as vector tiles).

- [website](#)
- [code](#)

SDM Explorer

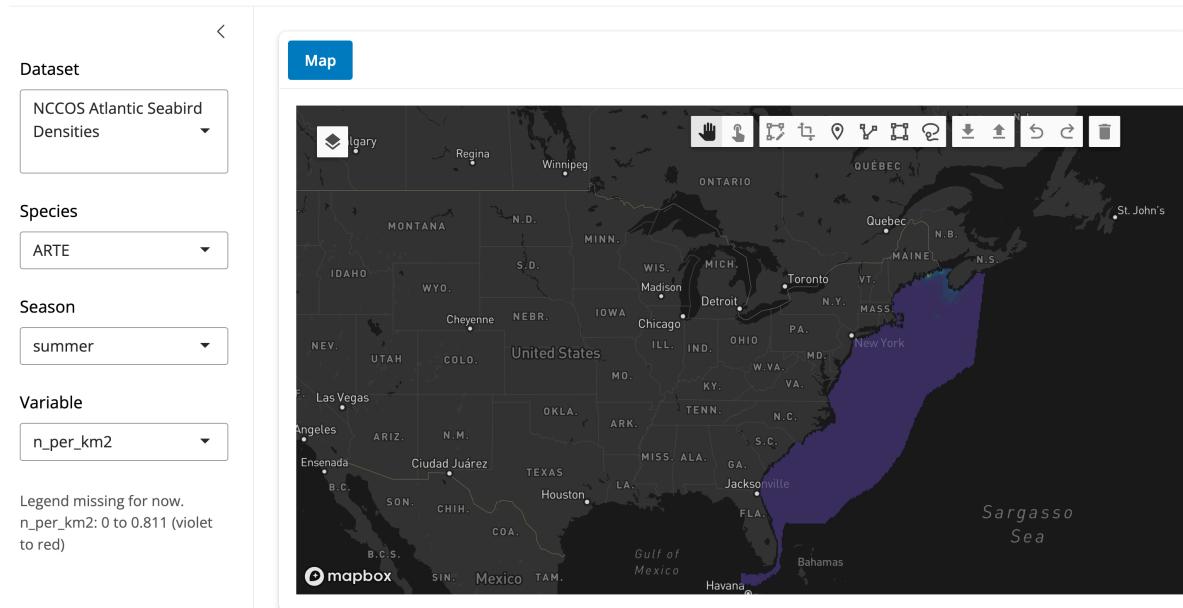


# Distributions, Raster

Show species distributions with high resolution rasters as cloud-optimized GeoTIFFs (COGs).

- [🌐 website](#)
- [📄 code](#)

SDM Raster Explorer



# Vulnerability Mapper

Combine species distribution models (raster) and vulnerability metrics (tables) to identify areas of high conservation concern.

- [!\[\]\(c5369b667fa38d2b0e143f5efeff87fa\_img.jpg\) website](#)
- [!\[\]\(794a2257640a72fd99b6bdb0e0c9dcce\_img.jpg\) code](#)

