

Ocean Superfarms: Finding the Best Spots to Grow Seafood

Vedika Shirtekar

11/28/25

Ocean Superfarms: Finding the Best Spots to Grow Seafood

Learning Outcomes

This assignment will reinforce key concepts in geospatial analysis by practicing the following:

- combining vector/raster data
- resampling raster data
- masking raster data
- map algebra

For this assignment, you are tasked with determining which Exclusive Economic Zones (EEZ) on the West Coast of the US are best suited to developing marine aquaculture for several species of oysters *and* a species of your choice. Suitable locations will be determined based on range of suitable sea surface temperature (SST) and depth values for the species.

Purpose

There is a growing need to sustain populations with food . One approach is harnessing marine aqualtculture. Aquaculture is defined as the (SOURCE). There is a great potential for aquaculture to offer benefits to supporting populations. In California, aquaculture is extremely diverse in production systems, cultured species, and products ().

Oysters and red abalone are two commercially important species of interest in California aquaculture. Oysters a have a defined range average SST of and depth of . Red abolne thrive in .

The purpose of this assignment is to evaluate the suitability of West Coast Exclusive Economic Zones (EEZs) for developing marine aquaculture for multiple oyster species and an additional selected species, red abalone. Suitable locations are determined based on species-specific ranges of sea surface temperature (SST) and depth. The assignment incorporates methods for working with both vector and raster data, including raster resampling and masking, as well as the application of map algebra techniques. The project consists of two components: (1) following a standardized workflow to create a final map of suitable oyster aquaculture areas in West Coast EEZs, and (2) developing a function that generates maps of suitability by EEZ based on user defined temperature ranges, depth limits, and species selection.

Part 1: Map of Suitable Locations for Oyster Aquaculture

Data Preparation

1. Load in the necessary packages.

```
# Import packages
library(here) # Load "here" to locate and reference files
library(tidyverse) # Load the tidyverse" for data cleaning
library(sf) # Load "sf" for GIS analysis
library(raster) # Load "raster" for accessing raster data types
library(ggplot2) # Load "ggplot2" for data visualization
library(tmap) # Load "tmap" for functions to create and layer maps
library(kableExtra) # Load "kableExtra" for table formatting
library(stars) # Load "stars" for integration with "sf"
library(terra) # Load "terra" for SpatVector and SpatRaster operations
```

2. Load in the spatial data:

2. Use `vect()` to read in the West Coast EEZ shapefile.
3. Use `rast()` to read in the bathymetry layer as a SpatRaster.
4. Create a list of the SST `tif` files using `list.files()`. Then, combine the SST rasters into a raster stack using `rast()`.

```
# West Coast EEZ
eez <- vect(here::here("data", "wc_regions_clean.shp"))

# Bathymetry raster
depth <- rast(here::here("data", "depth.tif"))
```

```

# Create a list of the tiff files for SST
sst_years <- list.files(path="data/", # File path
                        pattern = "average_annual", # File names matching pattern
                        full.names = TRUE) # Reference entire file names matching

# Stack all rasters
sst <- rast(sst_years)

```

3. Pass conditional checks to transform any mismatching CRS for a spatial object to the reference CRS (eez) using `st_crs()`.

```

# Create list of spatial objects
spatial_objects <- list(eez, depth, sst)

# Use eez's CRS as reference
ref_crs <- st_crs(spatial_objects$eez)

# Check and transform each tile with if/else statements
if (st_crs(spatial_objects$depth) != ref_crs) {
  warning("depth CRS does not match.
          Transforming to match eez CRS.")
  spatial_objects$depth <- st_transform(spatial_objects$depth, ref_crs)
} else {
  message("depth CRS already matches eez CRS.")
}

```

depth CRS already matches eez CRS.

```

# Check and transform each tile with if/else statements
if (st_crs(spatial_objects$sst) != ref_crs) {
  warning("sst CRS does not match.
          Transforming to match eez CRS.")
  spatial_objects$sst <- st_transform(spatial_objects$sst, ref_crs)
} else {
  message("sst CRS already matches eez CRS.")
}

```

sst CRS already matches eez CRS.

Data Processing

The following steps to process the SST and depth data are required prior to being combined.

3. Reproject the depth dataset to match the SST coordinate reference system (CRS) so the layers align spatially and can be analyzed together accurately. Then, create a single raster of average SST from 2008-2012 using `mean()`.

```
# Reproject to match sst CRS
depth <- project(depth, sst)

# Calculate average SST among all rasters
avg_sst <- mean(sst)
```

4. Subtract 273.15 from the single average SST raster (`avg_sst`) to convert average temperatures from Kelvin to Celsius.

```
# Update avg_sst in degrees Celsius
avg_sst <- avg_sst - 273.15
```

5. Ensure that both rasters match in terms of their CRS, resolutions, and extents prior to cropping.

```
# Do the CRS match?
print(paste("Do the CRS match:", crs(avg_sst) == crs(depth)))
```

```
[1] "Do the CRS match: TRUE"
```

```
# Do the resolutions match (require resampling)?
print(paste("Do the resolutions match:", res(avg_sst) == res(depth)))
```

```
[1] "Do the resolutions match: TRUE" "Do the resolutions match: TRUE"
```

```
# Do the extents match?
print(paste("Do the extents match:", ext(depth) == ext(avg_sst)))
```

```
[1] "Do the extents match: TRUE"
```

6. Crop the `depth` raster to match the extent of the `avg_sst` raster, then resample the cropped raster (`depth_sst_crop`) to match the resolution of `avg_sst` using the nearest neighbor method.

```
# Use crop() to crop depth to the extent of avg_sst
depth_sst_crop <- crop(depth, avg_sst)

# Resample with nearest neighbor method
depth_sst_crop <- resample(depth_sst_crop, avg_sst, method = "near")
```

Finding suitable locations

7. Reclassify `avg_sst` and `depth` using a defined reclassification matrix for the oyster-specific SST and depth range in `classify()`. The reclassification matrix should set suitable values to 1 and unsuitable values to 0.

```
# Preferred oyster range for SST: 11-30°C

# Define reclass matrix for un/suitable SST
reclass_matrix_sst <- matrix(
  c(-Inf, 11, 0, # Negative infinity (unbounded) to 11 degrees assigned 0
    11, 30, 1, # 11-30 degrees assigned 1
    30, Inf, 0), # 30 to infinity (unbounded) assigned 0
  ncol = 3, # Create three columns
  byrow = T # Fill by row
)

# Assign reclassified values to avg_sst
avg_sst_reclass <- classify(avg_sst, rcl = reclass_matrix_sst)

# Preferred oyster range for depth: 0-70 meters below sea level

# Define reclass matrix for un/suitable depth
reclass_matrix_depth <- matrix(
  c(-Inf, -70, 0, # -70 used for values below sea level
    -70, 0, 1, # Suitable values assigned 1 for -70-0
    0, Inf, 0), # > 0 assigned unsuitable (0)
  ncol = 3, # Create three columns
  byrow = T # Fill by row
)
```

```
# Assign reclassified values to depth
depth_reclass <- classify(depth_sst_crop, rcl = reclass_matrix_depth)

8. Identify suitable (value = 1) and unsuitable locations (value = 0) for both reclassified SST and depth by defining a function that multiplies two rasters. Then, stack the reclassified layers with lapp() and apply the function to generate a combined binary suitability raster.
```

```
# Create multiplication function to reference in lapp
multiply <- function(x,y){
  multi_raster <- x*y # Raster multiplication across all cells
  return(multi_raster)
}

# Return suitable (1) and unsuitable (0) cells
avg_sst_depth <- lapp(
  x = c(avg_sst_reclass,depth_reclass), # Stack rasters
  fun = multiply) # Apply multiplication function
```

Determine the most suitable EEZ

In order to rank zones by priority, it was important to determine the total suitable area within each EEZ. The total area of suitable locations within each EEZ was calculated using the following steps.

9. Project eez to match the CRS of avg_sst_depth.

```
# Project
eez <- project(eez, avg_sst_depth)
```

10. Use ifel() to identify suitable cells within the West Coast EEZs by replacing values of 0 with NA (unsuitable) and converting all remaining values to 1 (suitable).

```
# Select suitable areas
avg_sst_depth_suitable <- ifel(avg_sst_depth == 0,
  NA, # Replace with NA
  1) # Otherwise assign "1"
```

11. Calculate the total suitable aquaculture area (km^2) within each EEZ by masking, calculating cell areas, and summing suitable raster cells by region.

a. Rasterize each EEZ such that each pixel is labeled with its EEZ region. Rasterizing eez is crucial in this step because vector EEZ polygons are needed to operate in raster space so area can be summarized per region using raster-based functions (ex.zonal()).

```
# Rasterize eez regions
eez_rast <- rasterize(eez, avg_sst_depth_suitable,
                      field = "rgn") # By region
```

b. Create a mask to keep only raster values inside EEZ boundaries and remove everything outside by setting it to NA.

```
# Identify suitable cells in mask
suitable_cells_eez <- mask(avg_sst_depth_suitable, eez)
```

c. Create a raster using cellSize() where each cell contains its surface area in km^2 . Then, usezonal() to calculate the total suitable area within each EEZ by summing suitable raster cells areas by zone and joining the results to the eez_sf spatial data.

```
# Calculate cell areas ( $\text{km}^2$ )
cell_area <- cellSize(suitable_cells_eez, unit = "km")

# Convert to sf object
eez_sf <- st_as_sf(eez) # To have geometry

area_eez <- zonal(cell_area * suitable_cells_eez, # Identify rea of suitable locations onl
                   eez_rast, # Rasterized eez
                   fun = "sum", na.rm = T) %>% # Sum areas of all cells within each EEZ zo
rename(suitable_area_km2 = area) %>% # Rename for naming conventions
as.data.frame() %>% # Convert to data frame
left_join(eez_sf, by = "rgn") # Join on region

# Convert back into sf so eez data (includes calculated suitable area) for mapping
area_eez <- area_eez %>% st_as_sf()
```

12. Use tmap functions to create a map of suitable area within each EEZ.

```

# Create map
tm_shape(area_eez) +
  tm_polygons(
    "suitable_area_km2", # Color by suitable_area_km2 variable
    palette = "-mako", # Reverse blue scale
    style = "cont", # Continuous scale (styles referenced here: https://r-tmap.github.io/t
    title = "Suitable Area (km2)" # Rename legend title
  ) +
  tm_text("rgn", # Label by region
    size = 0.5, # Adjust size
    col = "white", # Adjust text color
    fontface = "bold", # Labels are bolded
    xmod = -.5) + # Adjust .5 from the left

  tm_layout( # Center title outside bounding box
    main.title = "Marine Aquaculture Suitability for Oysters in West Coast EEZs",
    main.title.size = 1.5, # Adjust title size
    legend.outside = TRUE, # Place legend outside map frame
    legend.outside.position = "right", # Place legend to right
    component.autoscale = FALSE, # Disable autoscaling for title
    outer.margins = c(0.01, 0.25, 0.01, 0.05) # Manually adjust map frame
  ) +
  tm_scale_bar( # Add scale bar for scale
    position = c(-.01, 0.08), # Move 1% from left and 8% from bottom
    breaks = seq(0, 500, 150)) + # Establish scale bar ranges

  tm_compass( # Add compass for orientation
    type = "4star",
    position = c("right", "top")) + # Adjust position
  tm_basemap("Esri.OceanBasemap") # Ocean basemap

```

13. Create a table using `kableExtra` to display both the total suitable area and the proportion of each EEZ that is suitable for oyster aquaculture, enabling comparison across regions independent of EEZ size.

```

# Table with kableextra for prop of suitable areas to EEZ area
area_eez %>%
  st_drop_geometry %>% # Drop geometry
  dplyr::select(region = rgn, # Select region

```

```

suitable_area_km2, # Select suitable area
total_area_km2 = area_km2 # Rename to total_area
) %>%
# Update suitable_area to be rounded to nearest hundredth
mutate(suitable_area_km2 = round(suitable_area_km2, 2),
       # Create new variable for prop of suitable area to total EEZ area
       percent_suitable = round((suitable_area_km2 / total_area_km2) * 100,
                                  1)) %>% # Round to nearest tenth

dplyr::select(-total_area_km2) %>% # Deselect total_area
# Enable title
kable(caption = "Amount of Suitable Areas by EEZ for Oyster Preferences") %>%
# Allow table to be striped with highlight option
kable_styling(bootstrap_options = c("striped", "hover"),
              full_width = FALSE, # Disable full width
              position = "center") # Center labels

```

Table 1: Amount of Suitable Areas by EEZ for Oyster Preferences

region	suitable_area_km2	percent_suitable
Central California	3656.82	1.8
Northern California	194.13	0.1
Oregon	1028.90	0.6
Southern California	3062.20	1.5
Washington	2435.93	3.6

Part 2: Generalized Function of Aquaculture for Species Preferences

The `species_preference` function applies the generalized workflow (Part 1) to streamline identification of suitable zones based on species-specific sea surface temperature (SST) and depth ranges for any species of interest. The function takes minimum and maximum SST values, minimum and maximum depth limits, and a species name as inputs and returns a map of EEZ regions shaded by total suitable area.

13. Verify coordinate reference systems (CRS) and load required spatial datasets before creating and running the function to ensure all spatial objects are correctly prepared for processing.

```
# Load in data again
```

```

# West Coast EEZ
eez <- vect(here::here("data", "wc_regions_clean.shp"))

# Bathymetry raster
depth <- rast(here::here("data", "depth.tif"))

# Create a list of the tiff files for SST
sst_years <- list.files(path="data/", # File path
                         pattern = "average_annual", # File names matching pattern
                         full.names = TRUE) # Reference entire file names matching
# Stack all rasters (stack() used earlier)
sst <- rast(sst_years)

# Create list of spatial objects
spatial_objects <- list(eez, depth, sst)

# Use eez's CRS as reference
ref_crs <- st_crs(spatial_objects$eez)

# Check and transform each tile with if/else statements
if (st_crs(spatial_objects$depth) != ref_crs) {
  warning("depth CRS does not match. Transforming to match eez CRS.")
  spatial_objects$depth <- st_transform(spatial_objects$depth, ref_crs)
} else {
  message("depth CRS already matches eez CRS.")
}

```

depth CRS already matches eez CRS.

```

# Check and transform each tile with if/else statements
if (st_crs(spatial_objects$sst) != ref_crs) {
  warning("sst CRS does not match. Transforming to match eez CRS.")
  spatial_objects$sst <- st_transform(spatial_objects$sst, ref_crs)
} else {
  message("sst CRS already matches eez CRS.")
}

```

sst CRS already matches eez CRS.

14. Define the `multiply` function to support raster multiplication within the `species_preference` function when used with `lapp()`.

```
# Define multiply function for global raster multiplication
multiply <- function(x,y){
  multi_raster <- x*y
  return(multi_raster)
}
```

15. Create a function called `species_preference` that incorporates the Part 1 workflow to generate a map of suitable aquaculture areas within West Coast EEZs based on species-specific SST and depth parameters.

```
# This function takes arguments:
# minimum and maximum sea surface temperature
# minimum and maximum depth
# species name

species_preference <- function(min_temp, max_temp, min_depth, max_depth, species_name){

  #### Assume files have been loaded in already with matching CRS checks

  ####-----
  #### Data processing

  # Reproject to match sst CRS
  depth <- project(depth, sst)

  # Calculate average SST among all rasters
  avg_sst <- mean(sst)

  # Update avg_sst in degrees Celsius
  avg_sst <- avg_sst - 273.15

  # Use crop() to crop depth to the extent of avg_sst
  depth_sst_crop <- crop(depth, avg_sst)

  # Resample with nearest neighbor method
  depth_sst_crop <- resample(depth_sst_crop, avg_sst, method = "near")

  ####-----
  #### Find suitable locations
```

```

# Define reclass matrix for un/suitable SST
reclass_matrix_sst <- matrix(
  c(-Inf, min_temp, 0, # Negative infinity (unbounded) to min_temp
    min_temp, max_temp, 1, # min-max temp assigned 1
    max_temp, Inf, 0), # max_temp to infinity (unbounded) assigned 0
  ncol = 3, # Create three columns
  byrow = T # Fill by row
)

# Assign reclassified values to avg_sst
avg_sst_reclass <- classify(avg_sst, rcl = reclass_matrix_sst)

# Define reclass matrix for un/suitable depth
reclass_matrix_depth <- matrix(
  c(-Inf, min_depth, 0, # Negative infinity (unbounded) to min_depth (below sea level)
    min_depth, max_depth, 1, # min-max depth assigned 1
    max_depth, Inf, 0), # > max_depth assigned unsuitable (0)
  ncol = 3, # Create three columns
  byrow = T # Fill by row
)

# Assign reclassified values to depth
depth_reclass <- classify(depth_sst_crop, rcl = reclass_matrix_depth)

# Return suitable (1) and unsuitable (0) cells
avg_sst_depth <- lapp(
  x = c(avg_sst_reclass, depth_reclass), # Stack rasters
  fun = multiply) # Apply multiplication function

####_-----
#### Determine most suitable locations within EEZs

# Project
eez <- project(eez, avg_sst_depth)

# Select suitable areas
avg_sst_depth_suitable <- ifel(avg_sst_depth == 0,
  NA, # Replace with NA
  1) # Otherwise assign "1"

```

```

# Rasterize eez regions
eez_rast <- rasterize(eez, avg_sst_depth_suitable,
                      field = "rgn") # By region

# Identify suitable cells in mask
suitable_cells_eez <- mask(avg_sst_depth_suitable, eez)

# Calculate cell areas (km^2)
cell_area <- cellSize(suitable_cells_eez, unit = "km")

# Convert to sf object
eez_sf <- st_as_sf(eez) # To have geometry

area_eez <- zonal(cell_area * suitable_cells_eez, # Identify rea of suitable locations on
                   eez_rast, # Rasterized eez
                   fun = "sum", na.rm = T) %>% # Sum areas of all cells within each EEZ zo
rename(suitable_area_km2 = area) %>% # Rename for naming conventions
as.data.frame() %>% # Convert to data frame
left_join(eez_sf, by = "rgn") # Join on region

# Convert back into sf so eez data (includes calculated suitable area) for mapping
area_eez <- area_eez %>% st_as_sf()

#####
##### Create map of suitable areas within EEZs
tm_shape(area_eez) +
  tm_polygons(
    "suitable_area_km2", # Color by suitable_area_km2 variable
    palette = "-mako", # Reverse blue scale
    style = "cont", # Continuous scale
    title = "Suitable Area (km2)" # Rename legend title
  ) +
  tm_text("rgn", # Label by region
          size = 0.5, # Adjust size
          col = "white", # Adjust text color
          fontface = "bold", # Labels are bolded
          xmod = -.5) + # Adjust .5 from the left

  tm_layout( # Center title outside bounding box
    main.title = paste("Marine Aquaculture Suitability for",

```

```

        species_name, # Include species name in title
        "in West Coast EEZs"),
main.title.size = 1.5, # Adjust title size
legend.outside = TRUE, # Place legend outside map frame
legend.outside.position = "right", # Place legend to right
component.autoscale = FALSE, # Disable autoscaling for title
outer.margins = c(0.01, 0.25, 0.01, 0.05) # Manually adjust map frame
) +
tm_scale_bar( # Add scale bar for scale
position = c(-.01, 0.08), # Move 1% from left and 8% from bottom
breaks = seq(0, 500, 150)) + # Establish scale bar ranges

tm_compass( # Add compass for orientation
type = "4star",
position = c("right", "top")) + # Adjust position
tm_basemap("Esri.OceanBasemap") # Ocean basemap
}

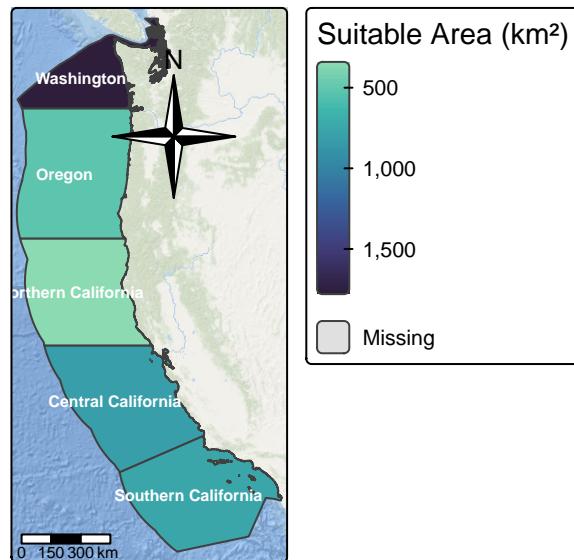
# Output:
# Map of EEZ regions colored by amount of suitable area

# Call function
species_pref_map <- species_preference(min_temp = 8, max_temp = 18, min_depth = -25, max_d

species_pref_map

```

Aquaculture Suitability for Red Abalone in West Coast



```
# Save finalized map to figs
tmap_save(species_pref_map, filename = "figs/species_pref_map.png", width = 8, height = 10)
```

Marine Aquaculture Suitability for Red Abalone in West Coast EEZs

