Lab 7 Hotspots

Sara Orofino

5/22/2019

Packages:

```
# Load packages:

library(raster)
```

```
## Warning: package 'raster' was built under R version 3.5.2
```

```
library(rgdal)
library(rasterVis)
library(maps)
library(rgeos)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
library(RColorBrewer)
```

Research Questions

1. What is the spatial distribution of global fishing and land-based threats in the CA Current?

2. Where were MPAs placed? Which threats do they address?

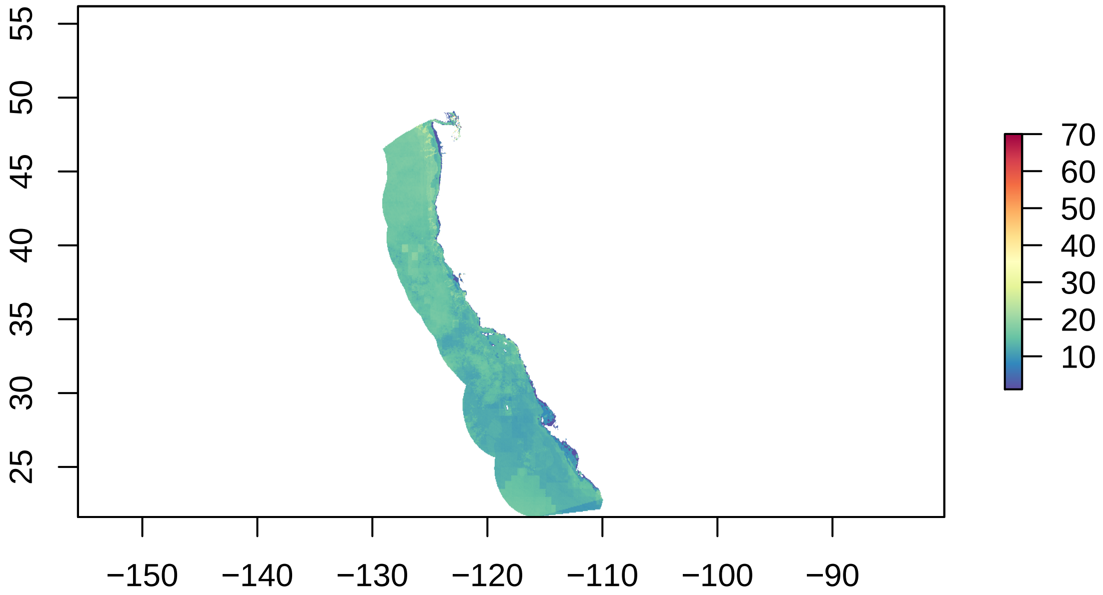
Part 1

Set color scheme and import raster data:

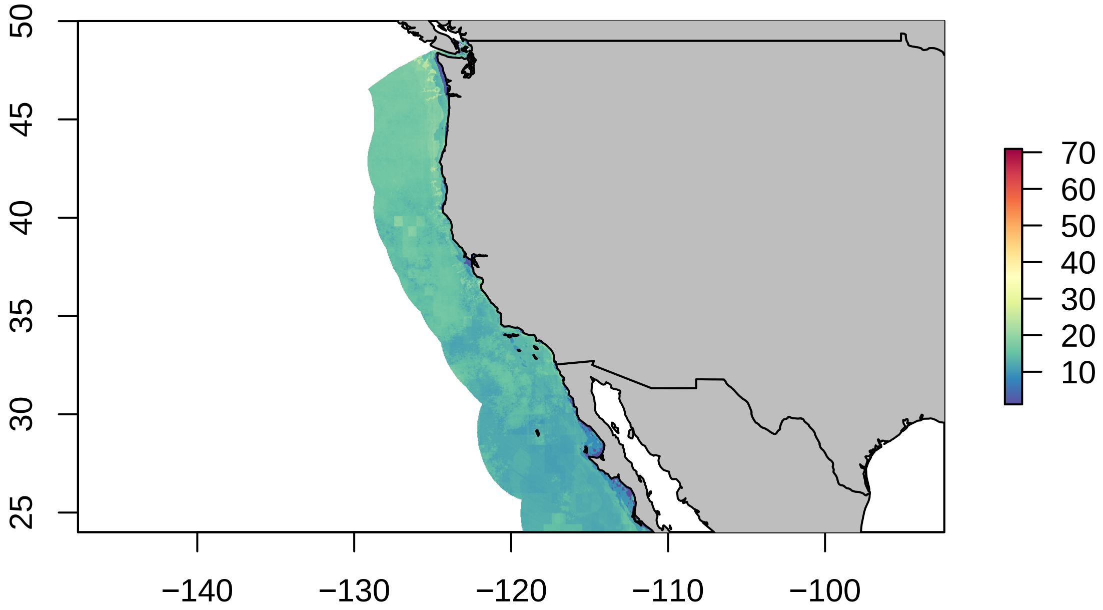
```
# Rainbow colors

cols = rev(colorRampPalette(brewer.pal(11, 'Spectral'))(255))

# set smaller margins for plotting
par(mar=c(2,2,1,1))

# Import raster data

all_threats <- raster("/Users/saraorofino/Documents/GitHub/CPLab7_Hotspots/Threats_data/full_modelnv.ti
```

Visualize:

```
# First plot just the CA current data:

plot(all_threats, col=cols)
```



```
# Add a basemap using the map package:
# ext=extent() is giving you the window you're viewing in this case lat/long
# add=T tells R to add it to the existing plot, fill and col are telling R to fill it in and with what

plot(all_threats, ext=extent(-130,-110,24,50), col=cols)
map('world', fill=T, add=T, col='gray')
```



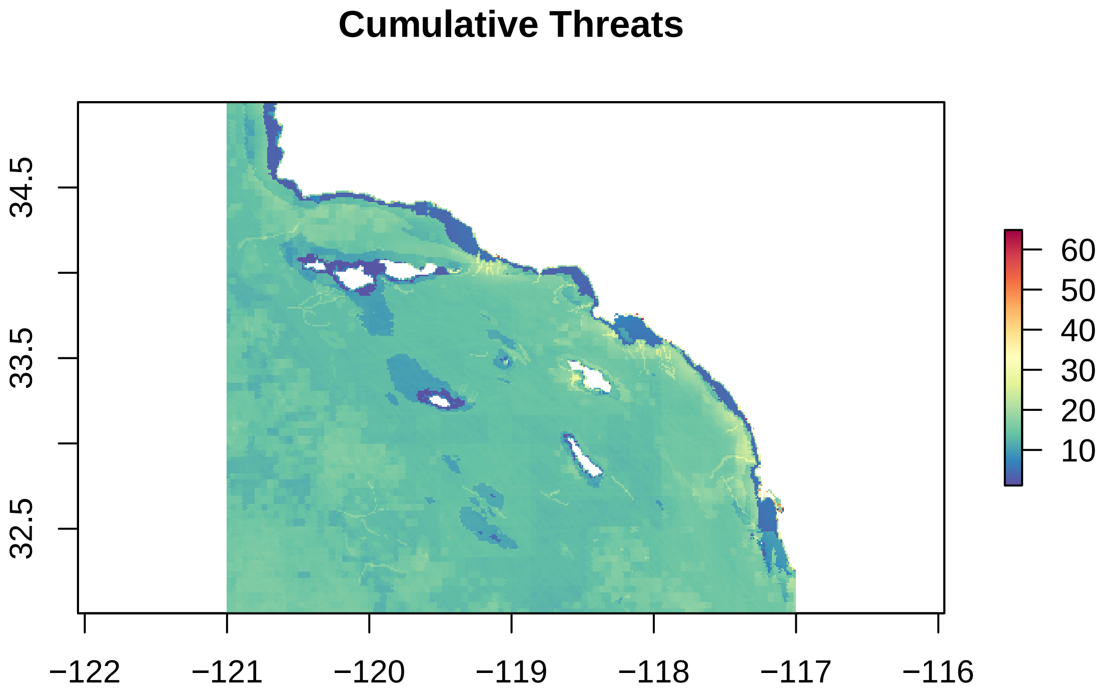
Extent and Zoom:

In the console if you type the raster layer you see:
- Class: what type of data it is (rasterlayer)
- Dimensions: Gives #rows, #columns, and total number of cells (rows * columns) - Resolution: resolution of each of the pixels (in the coorrdinate reference line - in this case lat/long)
- Extent: minimum and maximum of data set in units of the data (in this case lat/long)
- Coord. Ref: projection and datum, where it starts
- Data Source: where the file is located

- Names: what you named it when you imported it

Visualize a different extent, or subset, of the data:

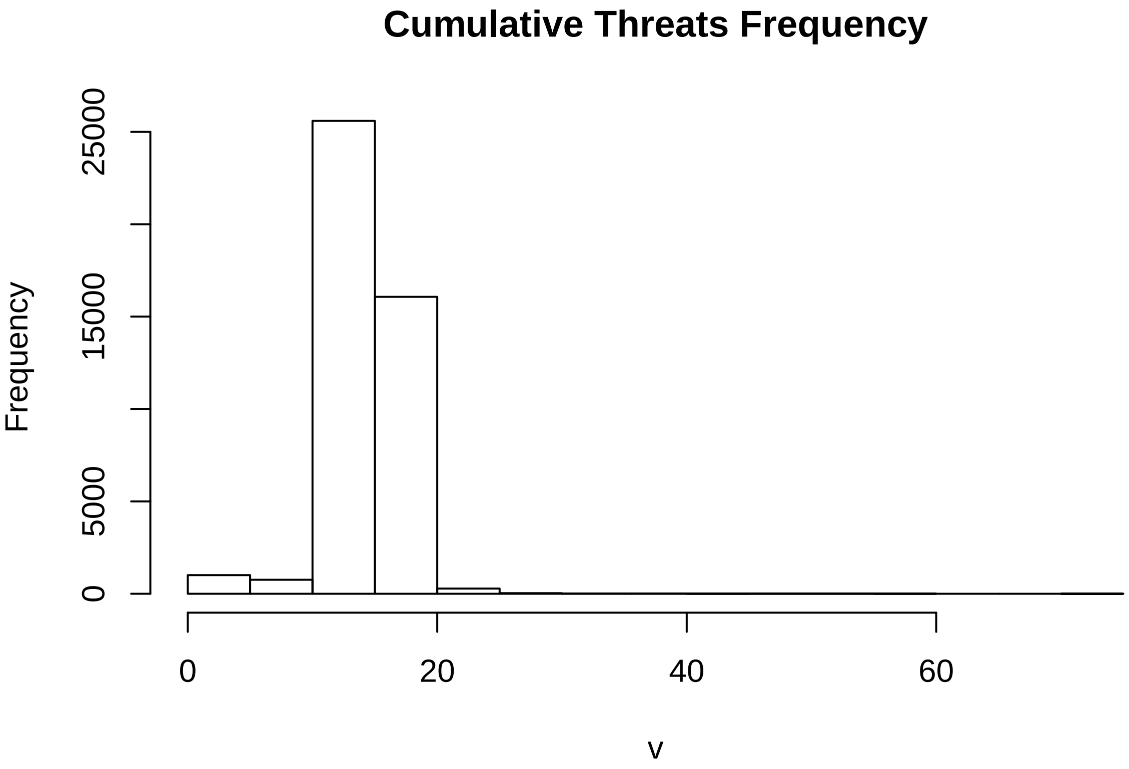
```
# Plot a good extent for Santa Barbara Channel
# Add a title using main

plot(all_threats, col=cols, ext=extent(-121,-117,32,35), main = "Cumulative Threats")
```



```
# Can use the zoom() function to have an interactive where you can zoom in by clicking
# NOTE - this doesn't work with pdf markdowns
```

Look at the distribution of the data with a histogram:

```
# Histogram to see the shape of data distribution

hist(all_threats, main = "Cumulative Threats Frequency")
```



```
# Note there are like 16,000 cells in this data so the histogram only shows a subset of about 1000

# cellStats() allows us to run basic statistics

cellStats(all_threats, mean) #mean = 14.00092
```

```
## [1] 14.00092
```

```
cellStats(all_threats, sd) #sd = 2.9589
```

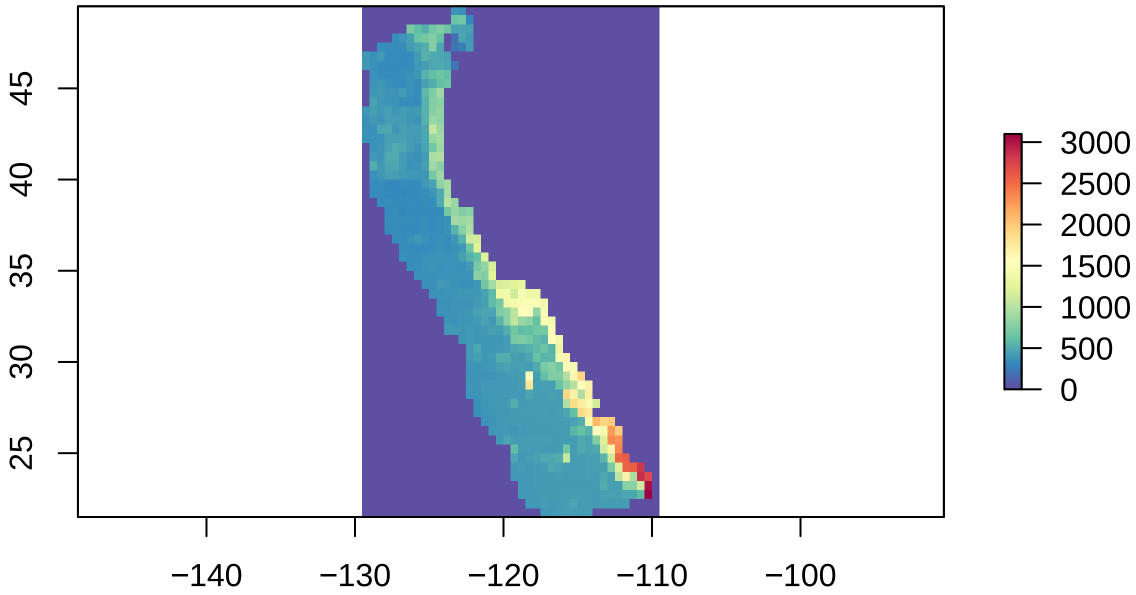
```
## [1] 2.958941
```

Part 2 Raster Calculations

Task: Overlay the top 20% of cumulative threats with the top 20% of species richness to find threat hotspots.
This requires four steps:
1. Import the data (already done for threats)
2. Resample the data to the same resolution
3. Reclassify the data to find the top 20%
4. Overlay the data and find hotspots

Step 1. Import the data

```
#Pull species data file:
all_spp <- raster("/Users/saraorofino/Documents/GitHub/CPLab7_Hotspots/ca_curr_sp_rich.tif")

# Visualize
plot(all_spp, col=cols)
```



```
# Notice the difference in resolution - larger resolution looks more pixelated than the threats data
```

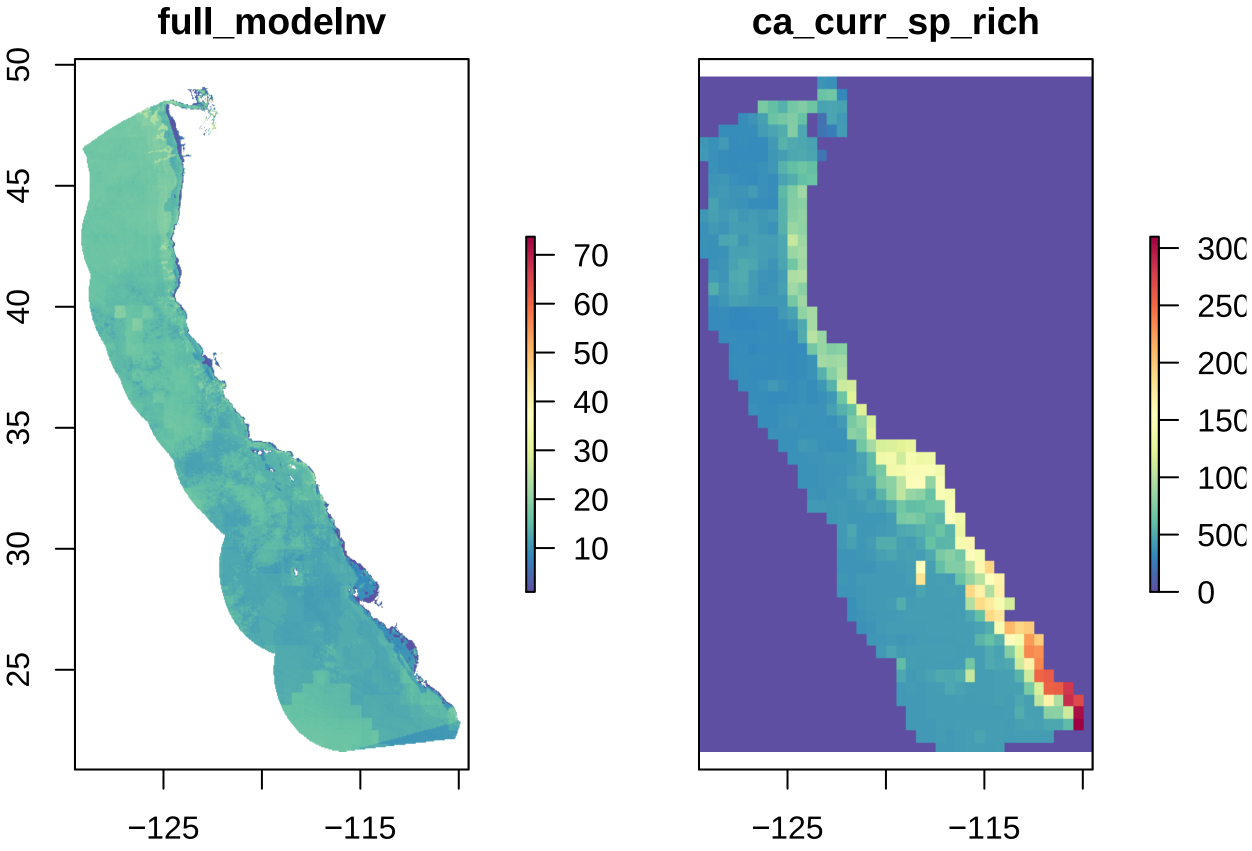
Note: calling the file in the console shows the extent and resolution are different than the threats layer - makes raster calculations impossible

Step 2. Resample the data

```
# Use crop() to change the extent - threats layer has the larger extent so we want to crop it to the sa

threats_crop <- crop(all_threats, all_spp)


# Use resample() to make the resolutions the same - resample the spp layer based on the cropped threats
# method = 'ngb' specifies we want to use a nearest neighbor algorithm to resample (instead of interpol
# Note: default for method is bilinear, need to specify for other methods
# progress = 'text' will show the progress in the console of longer running functions --> great tool

spp_res <- resample(all_spp, threats_crop, method = 'ngb', progress = 'text')
```

```
##
  |
  |                                                                 |   0%
  |
  |===========                                                      |  17%
  |
  |======================                                           |  33%
  |
  |================================                                 |  50%
  |
  |===========================================                      |  67%
  |
  |======================================================           |  83%
  |
  |=================================================================| 100%
##
```

```
# Use stack() to create a RasterStack object - checks that the layers will line up decently
spp_threat_stack <- stack(threats_crop, spp_res)
plot(spp_threat_stack, col=cols)
```



Step 3. Reclassify

We want to find the top 20% of cells in each layer and then combine the two layers to produce hotspots. We can do this using the reclassify() function in the raster package.

Function reclassify()
rcl argument - matrix for reclassification; requires 3 columns
- First two columns are the "from" and "to" of the input values
- Third column is "becomes"; the new value for your range
include.lowest argument - should the lowest value in the rcl be included?
- Default is false
right arguement - should intervals be closed on the right (and open on the left) or vice versa
- Default is true
- right = NA will leave both intervals open