Exercise 2: Choosing a Study Region

Adam B. Smith, Danielle Christianson, & Camilo Sanín

July 18, 2017

In this exercise you learn how to choose a *study region*, the areal extent of the place you're interested in. Later we'll modify this study region to meet certain objectives. In general there are three kinds of areas you'll need to (or would like to) define when modeling the niche/distribution of a species:

1. The current **range** of the species. This is the goal of "distribution" modeling (versus "niche" modeling). Hopefully the model will be able to provide you this.
2. A **background region** that encompasses the range *plus* surrounding locations from which you draw background data.
3. A **projection region** to which to project the model. Sometimes this is the same as #1 or #2, but if you're projecting model under climate scenarios, then the climate that the species currently prefers may move to locations very far away from the current range of the species. As a result, the projection region can be much larger than #1 or #2. Hence unless you have an a priori projection region in which you're interested, it's easiest to start by defining a projection region that encompasses #2 and #1 then work backwards from there.

**Take home**: Selecting a study region is a very visual task--I would be very cautious to automate the process without inspecting the results. I honestly almost never do this step in R because it's a lot easier to load maps and zoom around in a dedicated GIS program like ArcMap.

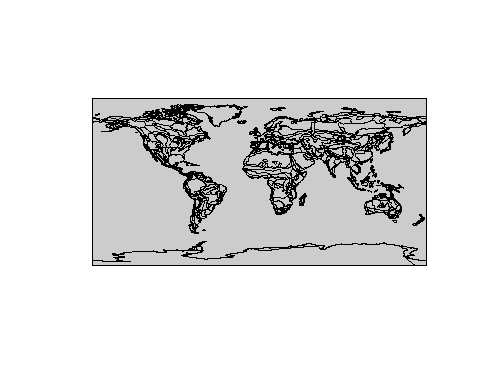
Nevertheless, we'll use R here because you may not have access to the pricey ArcMap and because it will teach us how to do some useful GIS operations in R.

We will delineate our study area using ecoregions, specifically the US Forestry Service's [Ecoregions of the Continents](http://www.fs.fed.us/rm/ecoregions/products/map-ecoregions-continents/) coverage. To speed things up, we've already downloaded this into the folder Ecoregions of the Continents (USDA). We'll be using this set of ecoregion simply because the file size is small which will help ensure subsequent operations run quickly. Note that there are other ecoregion definitions. You can find an incomplete catalog of them at [www.earthskysea.org/regionalizations](http://www.earthskysea.org/regionalizations/).

# load ecoregions  
eco <- rgdal::readOGR('./Ecoregions of the Continents (USDA)', 'provinces')

## OGR data source with driver: ESRI Shapefile   
## Source: "./Ecoregions of the Continents (USDA)", layer: "provinces"  
## with 636 features  
## It has 12 fields

sp::plot(eco, col='gray80')



(The double colons as in sp::plot specify that R should use the plot() function that's in the sp package. This is useful if functions from different packages have the same names.)

This is a shapefile, but R also treats it as if it were a data frame, and you can perform most many operations particular to data frames with it.

head(eco)

## AREA PERIMETER BEC\_ BEC\_ID DOM\_NUM DOM\_DESC DIV\_NUM  
## 0 12.436240 36.853440 4 4 100 POLAR DOMAIN 110  
## 1 2.487334 9.950983 8 15 100 POLAR DOMAIN M120  
## 2 22.279360 54.750890 9 31 100 POLAR DOMAIN M120  
## 3 1.518364 7.612691 10 13 100 POLAR DOMAIN M120  
## 4 8.819071 32.336720 11 20 100 POLAR DOMAIN M120  
## 5 4.097622 9.709030 13 9 100 POLAR DOMAIN 110  
## DIV\_DESC PRO\_NUM PRO\_DESC AREA.1 LEN  
## 0 ICECAP DIVISION <NA> <NA> 12.436235 36.853436  
## 1 TUNDRA REGIME MOUNTAINS M122 POLAR DESERT 2.487335 9.950983  
## 2 TUNDRA REGIME MOUNTAINS M122 POLAR DESERT 22.279363 54.750885  
## 3 TUNDRA REGIME MOUNTAINS M122 POLAR DESERT 1.518364 7.612691  
## 4 TUNDRA REGIME MOUNTAINS M122 POLAR DESERT 8.819072 32.336718  
## 5 ICECAP DIVISION <NA> <NA> 4.097623 9.709030

nrow(eco)

## [1] 636

The ocean is included as an ecoregion. Let's remove it... this will make life easier later and avoid the situation of choosing the ocean as a region that's potentially legitimate habitat for our species.

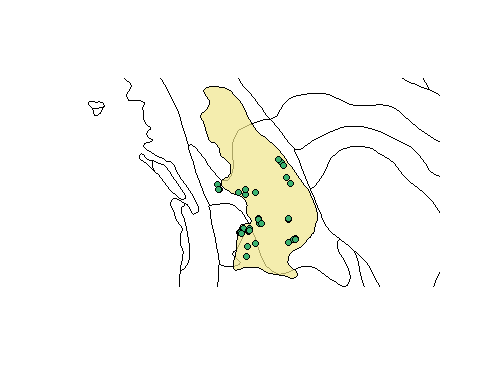
# Note which "domains" (uber-ecoregions) exist.  
unique(eco$DOM\_DESC)

## [1] POLAR DOMAIN HUMID TEMPERATE DOMAIN <NA>   
## [4] DRY DOMAIN HUMID TROPICAL DOMAIN outside polygon   
## 5 Levels: DRY DOMAIN HUMID TEMPERATE DOMAIN ... POLAR DOMAIN

# Remove ocean.  
eco <- eco[-which(eco$DOM\_DESC=='outside polygon'), ]

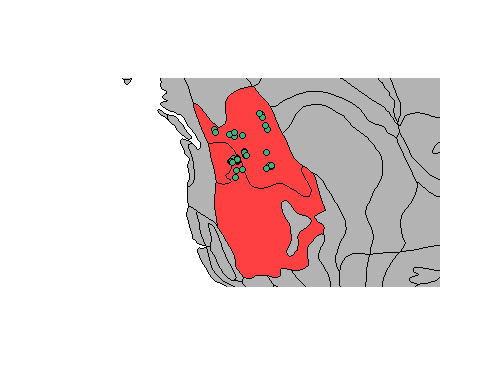
Now let's plot our area of interest.

# coerce species' records into a spatial (shapefile) points object  
recordsSpatial <- SpatialPointsDataFrame(  
 coords=cbind(records$longitude, records$latitude),  
 data=records,  
 proj4string=CRS('+proj=longlat +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0')  
)  
  
# plot  
plot(rangeMap) # set plotting extent  
sp::plot(eco, add=TRUE)  
plot(rangeMap, add=TRUE, col=alpha('khaki', 0.7))  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')



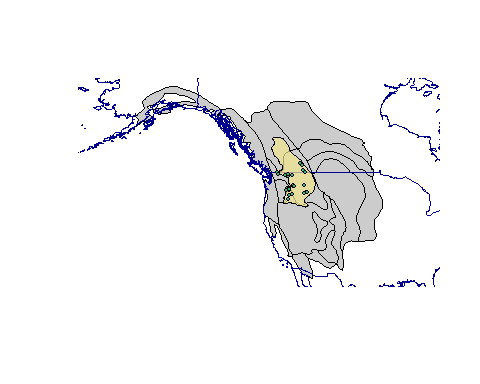
Now, to select our study region we'll start by choosing ecoregions which contain at least one record.

# select ecoregions in which there is at least one record  
ecoContain <- eco[recordsSpatial, ]  
  
# plot  
sp::plot(ecoContain)  
sp::plot(eco, add=TRUE, col='gray70')  
sp::plot(ecoContain, add=TRUE, col='brown1')  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')



This is nice, but some points lie right on the edge of an ecoregion, so it seems strange to exclude the adjacent regions. Let's include them, too.

# Get ecoregions that touch at least one other ecoregion that contains records.  
# This function returns a matrix, one row per ecoregion already selected  
# and one column per ecoregion that could be selected. Values indicate  
# if an ecoregion (column) touches an ecoregion (row) (1=touches, 0=  
# doesn't touch).  
touchMatrix <- gTouches(eco, ecoContain, byid=TRUE)  
  
# Get these ecoregions (ecoregions that have at least one "touch").  
touchVector <- colSums(touchMatrix)  
ecoStudyRegion <- eco[touchVector > 0, ]  
  
# Ensure regions containing points (not just touching containing regions)  
# are included... a hack because some regions represented more than once now  
ecoStudyRegion <- rbind(ecoStudyRegion, ecoContain, makeUniqueIDs=TRUE)  
  
sp::plot(ecoStudyRegion, col='gray80')  
sp::plot(countries, add=TRUE, col=NA, border='darkblue')  
plot(rangeMap, add=TRUE, col=alpha('khaki', 0.7))  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen', cex=0.5)



# save selected ecoregions for later use as a shapefile  
rgdal::writeOGR(  
 ecoStudyRegion,  
 './Terrestrial Ecoregions 2009 (TNC)',  
 'tnc\_terr\_ecoregions\_containOrAdjacentToContainingRegions',  
 driver='ESRI Shapefile', overwrite\_layer=TRUE  
)

The ecoregions themselves make somewhat of a strange shape, but that's OK. Right now we're more interested in their bounding box which we'll use to clip rasters and focus maps for plotting. Let's save the extent of the ecoregions in a folder to use for later.

studyExtent <- extent(ecoStudyRegion)  
dir.create('./Study Region', recursive=TRUE, showWarnings=FALSE)  
save(studyExtent, file='./Study Region/Study Region Extent.Rdata', compress=TRUE)

That's it!

# Reflection

1. How does the study region compare to the range? Remember, for our purposes the study region isn't just the polygons we selected but the box around them that contains them.
2. How much of the range occupies the study region for the species? How well is the presumed range (the range polygon) covered by the presence records? What are the implications of incomplete coverage for modeling?