The problem!

Imagine you have this situation: you have several global raster files and a shapefile with a few areas (e.g. Natural Parks). You want to generate a raster file using as mask each of the polygons in the shapefile for each of the original rasters. So, if you have 5 global rasters and a shapefile with 10 polygons, the output of this would be 50 rasters (a smaller raster for each polygon, cut from each of the 5 larger rasters).
Well I had this problem, as you might have guessed. I looked for solutions, maybe there are some, but I could not find any! I'm sure there are solutions out there… but let me show you mine!

This is a simple task, but if it is not automated, it's a grueling one… As such, the natural step, for me, was using R to automate this work.

The solution!

First we need to load the raster package and the shapefile:

```
library(raster)
polygon_areas <- raster::shapefile("C:/yourshapefile.shp")
```

This is the code for the function I created, called **crop_save**:

```
crop_save <- function(origin_folder, pattern, destination_folder,
name_sub_folder, crop_areas, name_crop_areas){
  file_list <- list.files(path = origin_folder, pattern)
  #Create folder
  dir.create(paste0(destination_folder,"/",name_sub_folder))
  how_many_areas <- nrow(crop_areas)
  #Create raster stack
  raster_stack <- stack()
  #File paths
  paths1 <- paste0(origin_folder,file_list)
  #Load rasters to stack
  for(i in 1:length(file_list)){
    raster_stack <- stack(raster_stack, raster(paths1[i]))
  }
  names_list <-  eval(parse(text=name_crop_areas))
  numbers <- 1:length(names_list)
  names_list <- paste0(as.character(numbers),"_polygon_", names_list)
  polyRR_list <- list()
  for(x in 1: nrow(crop_areas)){
    pol1 <- assign(names_list[x],crop_areas[x,])
    polyRR_list[[x]] <- pol1
  }
  for(j in 1:nlayers(raster_stack)){
    dir.create(paste0(destination_folder,"/",name_sub_folder, "/",
names(raster_stack)[j]))
    for(k in 1:length(polyRR_list)){
      a<-crop(raster_stack[[j]], polyRR_list[[k]])
      a<-mask(a,polyRR_list[[k]], filename =
paste0(destination_folder,"/",name_sub_folder, "/",
names(raster_stack)[j], "/", "RR",polyRR_list[[k]]$Id, ".tif"))
    }
  }
}
```

The arguments for this function are:

origin_folder – Where the original rasters are saved.
pattern – This is a character string to identify raster files: in the folder were rasters are saved there are, generally, other files. This argument allows the selection of only rasters (e.g. tif files).
destination_folder – Folder where the otput folder will be created.
name_sub_folder – Name of the sub-folder to be created inside the destination folder. Inside this, a folder is created for each of the original rasters where the smaller rasters for each polygon are saved.
crop_areas – Areas to be used in the raster croping (a SpatialPolygonsDataFrame created by importing the shapefile into R).
name_crop_areas – Column of the SpatialPolygonsDataFrame with the unique names or codes for the regions.

An example (not run, you have to try this with your own rasters):

```
crop_save(origin_folder = "D:/THIS_FOLDER/"
          , pattern = ".tif
          , destination_folder = "C:/OUTPUT/"
          , name_sub_folder = "Cut_rasters"
          , crop_areas = polygon_areas
          , name_crop_areas = "polygon_areas$Id"
          )
```

I hope this is useful!