Making Maps with R

Posted on 18 September, 2012 by kimgilbert

First off, thanks to Tim and Jeremy for the invitation to write a guest post here on using R to make maps! As a brief introduction, my name is Kim Gilbert, and I am a Ph.D. student at the University of British Columbia working with Mike Whitlock. I am broadly interested in population genetics and population structure, and am currently studying local adaptation in a tree species. If you want to know more, check out my website, where I also have this tutorial as a .pdf presentation.

Okay, onward with R! And apologies in advance that the R code provided will not show color coded text (a limitation of wordpress), but I decided that it is more useful to leave as text in order to allow copying and pasting rather than insert screenshots that might look nicer but require retyping on your part.

In the field of molecular ecology we see many, many maps. Maps are useful visual tools, from displaying sample sites to performing spatial analyses. To begin, you may ask "why would I want to make a map with R?" There are many answers to this question. R is one of several methods you could choose to make a map. The other main option, or at least most well-known, is ArcGIS, a software suite made by ESRI that includes ArcMap. ArcGIS is great; it is super powerful and has a lot of amazing features, but is not free. It requires a license that many universities provide to their students, however even with this license, it is quite a large suite of programs to install on your local computer and can be frustrating to work with through a remote connection.

R is free, it is open source, and users are constantly contributing new packages and functions. There is always talk of the steep learning curve in R, but I think the ArcGIS learning curve is just as steep if not more so. After taking a semester-long course that only taught ArcMap, I still barely grazed the surface of the capabilities of the software. And because I am not making maps every day, I have quickly forgotten many of the functions within ArcMap. With R, I can write a well-annotated script and then come back to it months or years later and say "ah, yes, that's how I did this, let me do that now with my new data." If you are already an R user, then making maps will be quite simple and intuitive since many of the functions take the same or similar arguments as plotting other types of figures. Plus, you have all of R's online help which is an incredible wealth of knowledge and is how I've learned everything I'm about to tell you. And not to harp on it, but one last favorite reason for making maps in R is that it is a cinch to go back to my data file and add or edit a sample point, re-run my script, and bam: I have a new map that is identical to my old one plus my newly added or edited point. I'm done in seconds.

Make a Simple Map

Now that I have successfully convinced you that you want to use R to make your next map, I will show you how. To start, let's just make a blank map with none of our own data. For this you need two packages: 'maps' which contains the functions we will use and 'mapdata' which has some basic world map data. Then with one line of code, you can create a simple map, in this case, of Canada:

```r
library(maps)
library(mapdata)
map("worldHires","Canada”, xlim=c(-141,-53), ylim=c(40,85), col="gray90”, fill=TRUE)
```

The function ‘`map()`‘ is what is doing the work here. We tell it to plot a map of “Canada”, out of the database “worldHires”. Just like for any other plot in R, we can specify the range of what we want to see, where the extent of our map is in terms of latitude and longitude. It is important to note here that the x-axis is longitude and the y-axis is latitude, which may be counter-intuitive to some since we think in terms of x then y, but we talk of GPS coordinates in terms of lat. then long. Then we can specify a color if desired, and in this case I chose to fill the map with a shade of gray.

All right, so that was easy enough. You can play around with the extent of that map by changing the xlim and ylim, which will allow you to zoom in on parts of Canada. Also check out the maps and mapdata documentations to see what data is already available for mapping other regions.

Plotting GPS Data & Shapefiles

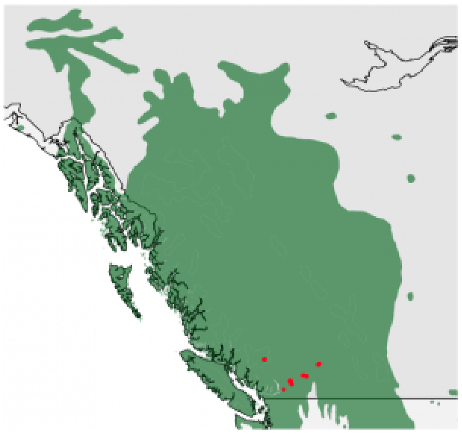
Now to add some of our own data to a map. A lot of the time maps in presentations simply plot sample sites which is very simple to do. All you need is a .csv file of all of your points consisting of at the bare minimum two columns: one for latitude, and one for longitude. Feel free to use my same FieldSamples.csv file to produce the map below. GPS points must be converted into **decimal degrees**. R will not read points in degrees, minutes, and seconds. For this example, I will add one other layer of data in addition to some of my own sample sites to show the species range.

My data for the species range is contained in what is called a shapefile. If you have never made a map before, I will quickly explain. A shapefile contains a layer of data that can be in the form of polygons, lines, or points, e.g. land zones, roads, or cities, respectively. (If you work with trees, the USGS has a great source of species ranges here; just watch for when they’ve last been updated.) A lot of GIS data is freely available online with a bit of searching. To read in data from a shapefile, we will make use of the ‘maptools‘ package, and depending on what type of data is contained within your shapefile will determine how it is read in. Polygon data uses ‘`readShapePoly()`‘, line data uses ‘`readShapeLines()`‘, and point data uses ‘`readShapePoints()`‘. After reading in a shapefile, it is added to my map by using the ‘`plot()`‘ function. Then for the GPS data, I read in my .csv file as I would for any other purpose, and I use the ‘`points()`‘ function to plot the points based on my two columns of longitude and latitude. I also load in a fourth package (‘scales‘) that I will explain below.

```r
library(maps)
library(mapdata)
library(maptools)  #for shapefiles
library(scales)  #for transparency
```

```
pcontorta <- readShapePoly("pinucont.shp")   #layer of data for species range
samps <- read.csv("FieldSamples.csv")   #my data for sampling sites, contains a column of "lat" and a column of "lon" with GPS points in decimal degrees
map("worldHires","Canada", xlim=c(-140,-110),ylim=c(48,64), col="gray90", fill=TRUE)  #plot the region of Canada I want
map("worldHires","usa", xlim=c(-140,-110),ylim=c(48,64), col="gray95", fill=TRUE, add=TRUE)  #add the adjacent parts of the US; can't forget my homeland
plot(pcontorta, add=TRUE, xlim=c(-140,-110),ylim=c(48,64), col=alpha("darkgreen", 0.6), border=FALSE)  #plot the species range
points(samps$lon, samps$lat, pch=19, col="red", cex=0.5)  #plot my sample sites
```

Details to note from this process: when creating the map, everything goes in order. I plotted the parts of Canada and the U.S. I wanted first, and this creates the full extent of my final map. I then plotted my species range on top of this map. This is where the 'scales' library came in handy, as it provides an easy way to make transparent colors. I wanted to preserve the coastlines and borders behind my species range, so using the function '`alpha("darkgreen", 0.6)`' tells R that I want this color to be 40% transparent (60% opaque). This could also easily be accomplished without transparency by plotting the countries again on top but with no fill. You can see that for all steps after the first, I have included '`add=TRUE`' because data are being added to the existing plot, and otherwise there would be no map upon which we are plotting the data. This is important to note because data are being plotted in layers, so things can be covered up by something that is plotted later down the line. Then lastly I plotted my GPS points. You can add some final touches such as drawing a box around the map with '`box()`', adding text with '`text()`' and specifying a GPS point as the x and y coordinates of where to place the text, or adding a scale bar with '`map.scale()`' which brings me to our next important topic.

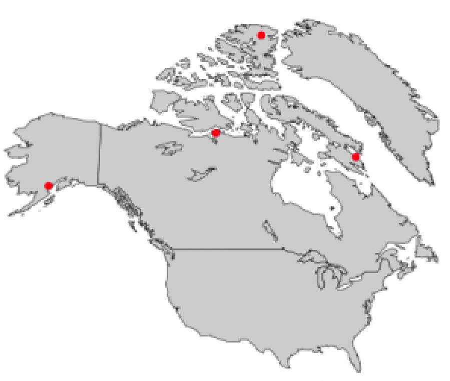


Projected Maps

When plotting a map such as the one above where I am showing quite a large area, and I am also looking at a more northerly region of the globe, there is distortion that occurs. As we all learned way back when, a globe is the most accurate representation of the Earth's surface. When we can't look at a globe, we can use map projections. On my unprojected map above, if I plot a scale bar, depending upon what spot on the map I choose for its location will determine what it conveys. If I plot it at the top, it makes Vancouver Island (the island at the bottom left of the map) appear to be much smaller than it actually is, and if I plot the scale at the bottom of the map, it makes the top of my map look like it is covering a much larger area than it actually is. Many thanks to J.S. Moore for pointing this out to me and helping to figure out plotting points on projected maps. There is a catch here, in that to the best of my knowledge it is still not possible to draw a map scale on a projected map (but if anyone does know, please share). Instead, when making a projected map, it is possible to draw grid lines on your map.

The package 'mapproj' allows the creation of projected maps, and it includes many different projections that can be chosen from to best suit your needs. In this example, we use the 'gilbert' projection:

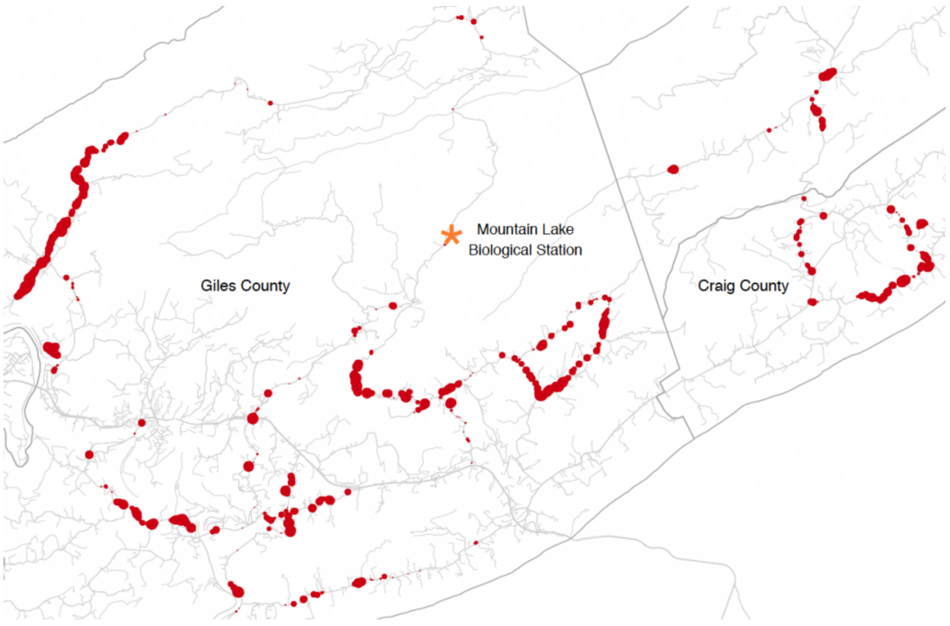
```
library(mapproj)
map(database= "world", ylim=c(45,90), xlim=c(-160,-50), col="grey80", fill=TRUE, projection="gilbert", orientation= c(90,0,225))
lon <- c(-72, -66, -107, -154)  #fake longitude vector
lat <- c(81.7, 64.6, 68.3, 60)  #fake latitude vector
coord <- mapproject(lon, lat, proj="gilbert", orientation=c(90, 0, 225))  #convert points to projected lat/long
points(coord, pch=20, cex=1.2, col="red")  #plot converted points
```



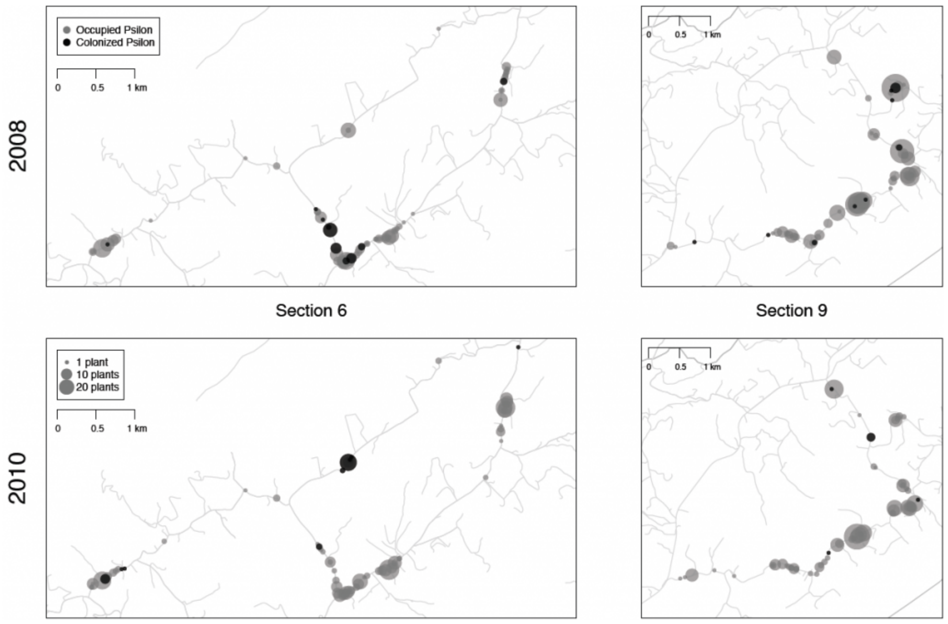
The ‘gilbert’ projection requires an additional parameter, ‘orientation’, telling R where the North Pole is located, which R needs in order to plot correctly. Extra parameters such as this will vary with the projection you choose. Once we’ve plotted our projected map, we want to add our points to it, but if we try this the way I just showed on the previous map, it will not work. The points must first be converted into the chosen projection, and then can be plotted on the map. This is done in the last two lines of code above, from the two vectors of fake points created in the preceding two lines. The function ‘`mapproject()`‘ takes our vectors of latitude and longitude and with the same projection information converts our points to the new object ‘coord’ which we then plot as we would normally. The function ‘`grid.lines()`‘ can then be used to add grid lines.

More R

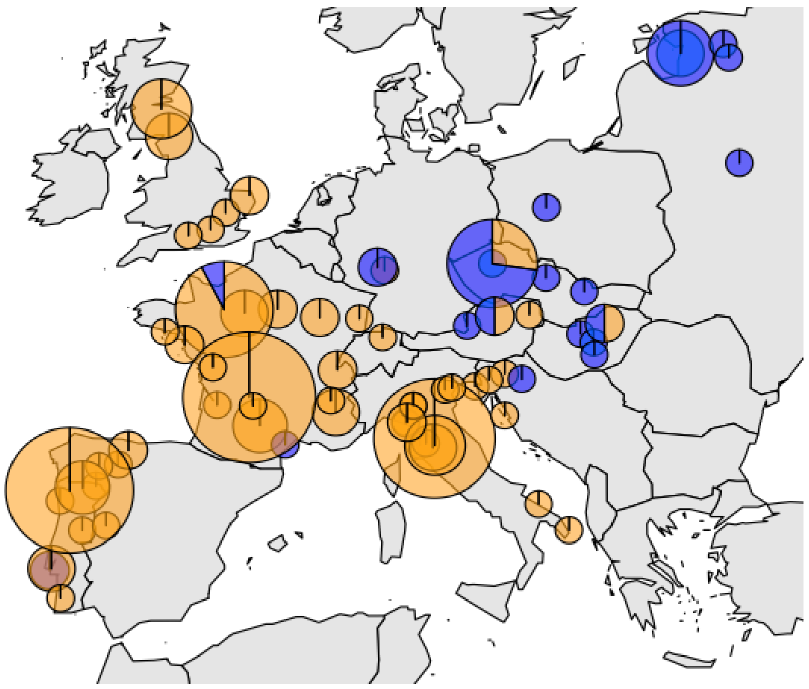
That covers the gist of basic map-making in R. There are many other packages for mapping or useful tools in conjunction with creating maps. Graduated symbols and colors can easily be made if you have a column of data associated with your GPS points by using that column within your ‘`cex`‘ or ‘`col`‘ specifications. A map legend is plotted the same as any other plot in R, except plotting location is given in terms of a GPS point on the map. Multiple maps can be arranged, for example to create insets, with the ‘`layout()`‘ function. And yes, you can also plot pie charts on a map! This is done through either the ‘plotrix’ package or the ‘mapplots’ package and using the functions ‘`add.pie()`‘ or ‘`floating.pie()`‘ respectively. And those familiar with R know that all of the images you create can be output easily to a file such as a .pdf by bracketing your code with ‘`pdf()`‘ and ‘`dev.off()`‘.



Displaying population sizes with graduated point sizes (cex=data$popsize)



Using the 'layout()' function to arrange maps in a .pdf
mat <- matrix(c(1,2,3,4),2, byrow=TRUE)
layout(mat, c(8.9782, 5.45), c(6,6))
par(mar=c(0.05,0.05,0.05,0.05))

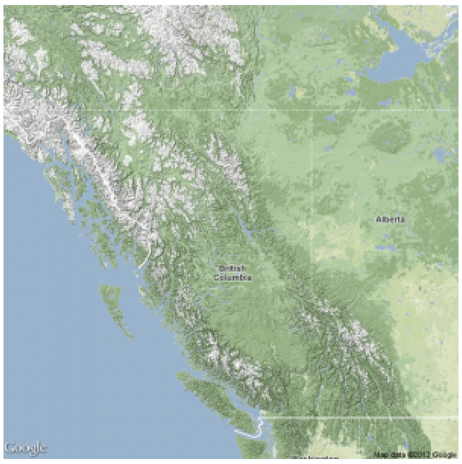


Plot pie charts on a map using 'add.pie()' in the 'mapplots' package.
add.pie(z=c(east, west), x=lon, y=lat, radius=sqrt(tot), col=c(alpha("orange", 0.6), alpha("blue", 0.6)), labels="")
#z indicates the portions of the pie charts filled by each given type, x & y are coordinates for the point, and radius is to designate the size of the circle for the pie chart
#to plot all points, I run a loop to run through my data one point at a time and make each pie chart; there are most likely more efficient methods

RgoogleMaps

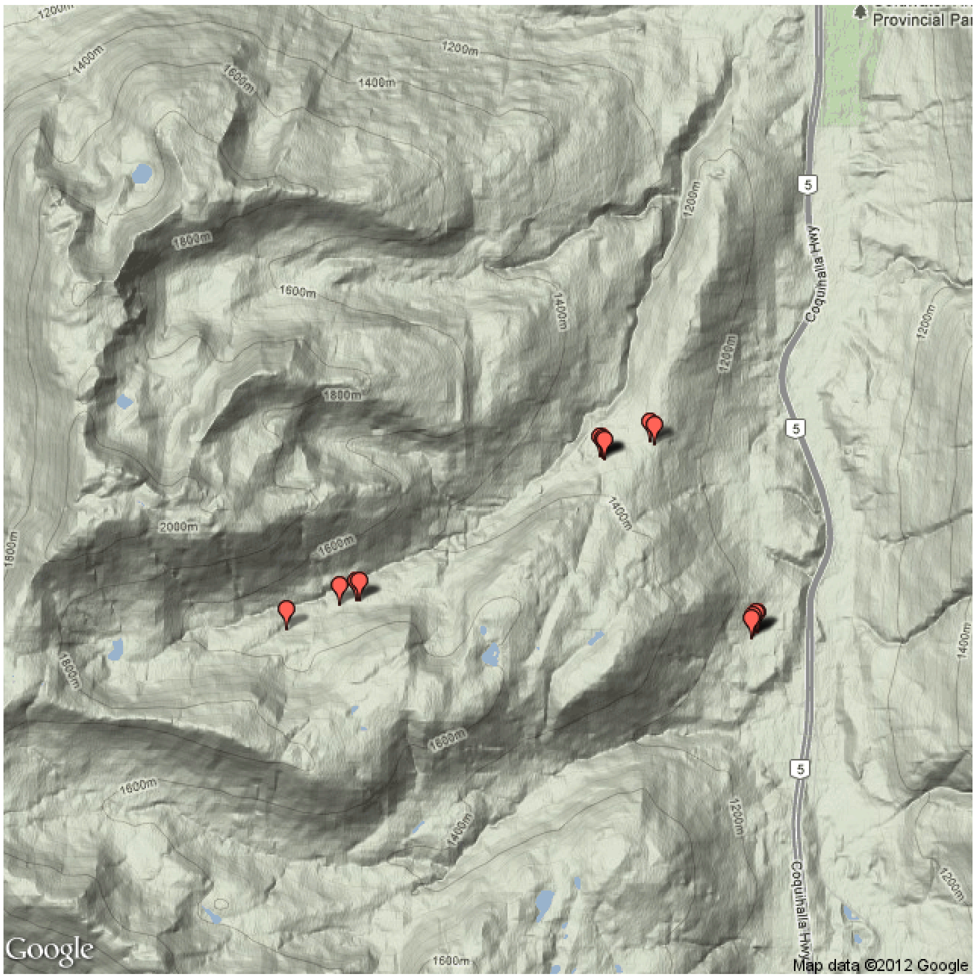
I will introduce one last topic before wrapping this post up because I think this is a useful package. 'RgoogleMaps' allows you to plot data points on any sort of map you can imagine seeing (terrain, satellite, hybrid) from using Google Maps in your browser. This can be useful if you want more than simply a blank map with points plotted on it. There are two basic functions: '`GetMap`' and '`GetMap.bbox`', where the latter takes care of some of the overhead in the former. With these functions, maps are plotted straight to an output file; they will not show up in your quartz window as before.

```
library(RgoogleMaps)
lat <- c(48,64) #define our map's ylim
lon <- c(-140,-110) #define our map's xlim
center = c(mean(lat), mean(lon))  #tell what point to center on
zoom <- 5  #zoom: 1 = furthest out (entire globe), larger numbers = closer in
terrmap <- GetMap(center=center, zoom=zoom, maptype= "terrain", destfile =
"terrain.png") #lots of visual options, just like google maps: maptype = c("roadmap",
"mobile", "satellite", "terrain", "hybrid", "mapmaker-roadmap", "mapmaker-hybrid")
```



The above code produces a terrain type map from Google, shown on the right. We can add points onto one of these maps, however it is not as straightforward as the maps made previously. Perhaps it is a limitation of my computer's memory, but I have also run into errors when attempting to plot more than about 35 data points. But for those who are interested, it would be worthwhile to look into these methods more in depth than I have. To plot points, we create a data frame to contain them in addition to anything that was initially read into R:

```
samps$size <- "small"  #create a column indicating size of marker
samps$col <- "red"   #create a column indicating color of marker
samps$char <- ""   #normal Google Maps pinpoints will be drawn
mymarkers <- cbind.data.frame(samps$lat, samps$lon, samps$size, samps$col, samps$char)
  #create the data frame by binding my data columns of GPS coordinates with the newly
created columns
names(mymarkers) <- c("lat", "lon", "size", "col", "char")  #assign column headings
lat <- c(48,60)  #now we are plotting the map
lon <- c(-140,-110)
terrain_close <- GetMap.bbox(lonR= range(lon), latR= range(lat), center= c(49.7,
-121.05), destfile= "terrclose.png", markers= mymarkers, zoom=13, maptype="terrain")
```



There are lots of other R packages and capabilities out there that I have not touched on here, many of which are listed in the .pdf file I mentioned at the start of this post. I hope these examples have been useful, and that many of you will now begin creating your maps in R and exploring all of the other mapping capabilities. Thanks for reading!

Recommend 731 Tweet 89 Share 36 [Share and Enjoy]

About kimgilbert

[Kim Gilbert] is a PhD candidate in the [Department of Zoology] at the University of British Columbia, and can also be found on twitter [@kj_gilbert].
[View all posts by kimgilbert →]

This entry was posted in howto, R, software and tagged map, R, spatial data. Bookmark the permalink.

- *Matt Pennell*
  Thanks Kim!! This is fantastic.

- *RJC*
  give ggmap a look too: https://sites.google.com/site/davidkahle/ggmap

- *Nick Sard*
  This is very cool! Thanks for the post!

- *http://cartodb.com Andrew Hill*
  Very nice stuff Kim! If you are also interested in publishing/sharing data online, you should take look at a recent post I wrote about mapping data in R from CartoDB, http://blog.cartodb.com/post/22719502434/cartodb-with-a-capital-r

- *kimgilbert*
  Thanks all, and thanks for the links, great stuff! The ability to do analyses of your spatial data with CartoDB looks really cool, too.

- Pingback: Friday Coffee Break « Nothing in biology makes sense!

- *http://www.datavoreconsulting.com Travis Hinkelman*
  Nice post! You can get syntax-highlighting capabilities by using the WP-Syntax plugin and adding `pre lang="rsplus"` to the HTML.

  - *http://www.denimandtweed.com/ jeremyyoder*
    Thanks! That should come in handy.

- *Mao*
  Thanks greatly, i really should learn this fantastic R.

- *Kristina Lemson*
  The tutorial is great! Any hints on sources of code for converting UTM coordinates to dec degrees, or packages that can use UTM straight (seems cleaner to me) ?

- *kimgilbert*
  Hi Kristina, I've never done this, so perhaps someone more knowledgeable might chime in, but looking around it seems possible to either convert or just use UTM coordinates. I think the packages 'sp' and 'rgdal' should have some useful things. There are transformations you can do with 'proj4string', and this .pdf might be useful to you as well as some of the code in these help chains here, here, and here.

- Pingback: Créer sa carte de metro avec les données de la ratp | Romain Rossier

- *migee*
  Great post!! Few questions though. How did you get/download those two maps showing counties (http://www.molecularecologist.com/wp-content/uploads/2012/09/MetapopBySizeFull.png)?

  And, how did you add the scale bar?

  Sorry if the questions are too novice, but i am just getting excited by this post.

  Thanks again.

- *kimgilbert*
  Hi Migee,

  For the county lines, I searched around online for the shapefiles. Those are in Virginia, so I think I may have found the data available for free from VADOT, at least for the roads. I think the ESRI website may have data for county lines, or the USGS. Just googling what you want is your best bet.

  If you check out the .pdf I link to at the top, slide 24 shows how you can import that data into R. Also check out slides 31 and 32 for the scale. It's the 'map.scale()' function in the 'maps' package.

  I hope that helps!
  Kim

- *Joanna Malukiewicz*
  Hi Migee,
  Thanks so much for this post! This has been a great introduction to map making in R. One of the maps I have to make is similar to the map you show above with the green species range and plotted GPS points. I was wondering how you made the shapefile (the pcontorta files) that shows the species range? Is that ultimately based on a csv file with geographical coordinates outlining the range? I guess I am wondering if you made that shapefile yourself and how did you do it? Thanks!

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*

Hi Joanna,

I did not make the shapefile shown, I posted the link to the website where I was able to get this one. There is a way to make shapefiles (perhaps even in R), but I do not know how. A GIS specialist is probably your best bet to learn that process. Good luck!

Kim

- *Joanna Malukiewicz*
  thanks!

  - *Joanna Malukiewicz*
    and sorry for not getting your name correctly. i did not mean to do that.

- *Alice Klar*
  maptools and or raster

  #make a shapefile

  dataShp=as.data.frame(myData)

  coordinates(dataShp)=c("X","Y")#this assumes your long and lat are labeled X and Y

  proj4string(dataShp)=CRS("+proj=longlat +datum=NAD83″)#change projection to fit your input, then can transfomr to whatever you like

- *Essellgee*
  I wanted to thank you as well for an excellent discussion. Using the stuff that's out in web for this can be challenging – many examples for maptools don't work because it appears a lot has been deprecated.

  For those of us following along at home, replicating your work, a copy of the FieldSamples.cvs might be helpful. I couldn't find one here.

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Sure thing! I added the file into the post, so you should be able to download it now. In the paragraph just before I discuss shapefiles.

- *Catherine Reeb*
  Thanks a lot! just what I was looking for 2 hours on the web, and your site is the best one, simple, clear.

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Thanks Catherine, glad to be of help!

  - *kimgilbert*
    Thanks Catherine, glad to be of help!

- *Susana*
  Thank you! it is very clear and useful! 🙂

- *Jocelyn*
  Hi Kim! How did you loop the add.pie? Awesome tutorial, by the way!

  - *kimgilbert*
    Hi Jocelyn, I looped through each row of data, with each iteration plotting one point from the data in that row with the add.pie function. I think an 'apply()' would also do the job. I'm actually working on putting something together to make this easier, so I'll post an update when that's ready.

- Pingback: Species distribution models in R | The Molecular Ecologist

- *Shane*
  Hi Kim. Great technology at use here. I'm a computer games programmer on an ecological project for my work placement. Is there any way to use this on a smaller scale? For example my job is to create a map of the local castle grounds and all of theworld renowned trees in it. So far I have created a map of the grounds on an x,y map but in order to add in custom points(I.e individual trees I have to manage the relation between gps and the x, y scale. Will this software be able to handle a custom map using coordinates I provide it?

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Hi Shane, if I understand what you mean correctly, then R should work for you just fine. When plotting your map, you will likely have very precise lat/long points with many digits since you'll be zooming in on a small scale, but I don't foresee any problems otherwise. You can just plot the trees you have by GPS points on top of that. You might want to look more into RgoogleMaps too since it could make the map more visually interesting on such a small scale. Good luck!

- *Jimmy*
  Very helpful tutorial. It's my understanding of map projections that only one dimension is distorted. That is, for a two dimensional projection of a sphere, degrees can remain consistent in one direction but not the other. If this is the case, a scale bar is indeed not possible, but perhaps depending on the intent of the map, there are other useful representations of scale. If linear distance is important, it could be represented across the top and bottom of the map. If area is important, a circle or square of standard area could be plotted at various points up the side of the map. In the case of the first map in your tutorial, we would see fewer kilometers stretched out across the top than at the bottom, and a series of circles or squares of increasing size up the vertical axis, each representing 10 or 100 km^2. Just a late night thought!

- Pingback: It's e-MER-ging…MERS-CoV Spread Map | Why We Are Screwed

- *Petar*
  Thanks, great work! 😃

  How did you make the colors look like that (fading)?

  When I copy this text
  col=c(alpha("orange", 0.6), alpha("blue", 0.6))
  I get: could not find function "alpha"
  could not find function "alpha"

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Hi Petar, the 'alpha()' function requires the 'scales' package, so install scales, load the library with "library(scales)" and then re-try your code.

- Pingback: Nuts and Bolts: Modern maps of Eurasia in R | Alien Plantation

- Pingback: Geodaten sichtbar machen: Gehversuche mit R | run the numbers

- *Jan*
  This post has changed my life! You have no idea how useful it has been. Thanks!

- Pingback: Mapas en R… los partidos que gobiernan los estados en México (o, mejor dicho, el PRI que gobierna casi todo México) | Chanchullo Político

- *Carlos Rodríguez-Saltos*
  Beautiful post!!, a favorite!, thanks!

- *Elizaveta Ershova*
  Thanks for this great tutorial. One question: do you know of any way to plot outside the (-180, 180) longitude range? For example, is there a way to plot a map centered on -180W (i.e. on the Bering Sea)?

- *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
  Hi Elizaveta, that is a great question and was fun to find an answer for!

  So in one way, you can plot outside of the -180 to 180 bounds, e.g. you could do xlim=c(-260,-150) and you map would center somewhere around the Bering Sea, however nothing west of -180 gets plotted. Try: map("worldHires", xlim=c(-260,-150),ylim=c(40,80)) You could do the reverse with xlim=c(150,260) with the same result. And I so far had trouble overlaying these as separate maps into one map.

  But, there is an easier solution that already exists. The maps package already has a Pacific centric set of map data that you can load in and use, and the mapdata package has the same but of higher resolution, and these plot from 0 to 360. Try both of these:

  ?world2MapEnv #the description of the data
  data(world2MapEnv) #load in Pacific centric map data, low resolution
  map('world2′, xlim = c(100, 300), ylim=c(40,80))
  map.axes() #see how it labels longitude

  data(world2HiresMapEnv) #load in map data, high res
  map('world2Hires', xlim = c(150, 230), ylim=c(45,75))
  map.axes() #see how it labels longitude (image attached)

  You could try a different approach and use a projection as well: map(projection="sp_mercator", wrap=TRUE, ylim=c(-60, 75), interior=FALSE, orientation=c(90, 180, 0)) #center on north pole, xlim=c(-180, 180)

  And lastly, the solution to this in RgoogleMaps is quite easy; you just change the point designated as the center of the map.

  library(RgoogleMaps)
  lat <- c(50,60)
  lon <- c(-180,180)
  center = c(mean(lat), -180) #tell what point to center on
  zoom <- 4 #zoom: 1 = furthest out (entire globe), larger numbers = closer in
  terrmap <- GetMap(center=center, zoom=zoom, maptype= "terrain", destfile = "terrain.png") #(image attached)

  I hope this helps and good luck!
  Kim

  - *Elizaveta Ershova*
    Wow, thanks for such a quick and useful reply! This is exactly what I was looking for.

    P.S. Rather than running a loop, to plot multiple pies you can use the draw.pie command. Super easy to use once you transform your data into the right format using "make.xyz".

    - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
      Great! Thanks for the tip on draw.pie I hadn't figured that one out yet!

- Pingback: Pinus contorta distribution map in #rstats | On the other hand

- *Carmen*
  Great! Thanks!

- Pingback: Haberlo sabido antes: consejos para biólogos que usen R o que quieran aprender | Más Ciencia por México

- *Sarah Supp*
  Hi Kim! Great tutorial! I found it very helpful to make maps of my data. I'm especially interested in plotting points on google maps. But I ran into the same problem you did "but I have also run into errors when attempting to plot more than about 35 data points". Have you found a solution for this? I have 46 points that I would like to put on a map of North America, and I can't seem to do it with this package. If I plot fewer, it seems to work fine, but once I hit 35, I just get a long list of errors.

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Hi Sarah, great glad you found it useful! Interesting, yeah I haven't gone back into RgoogleMaps for a while, but maybe it's time I try to see what's going on. Or if anyone else has any answers…? I'll let you know if I figure it out!

- *guest*
  The clearest and most useful explanation I have ever found of anything R! Thank you!

- *Pablo*
  This came so handy for me! thanks a lot. I have a couple of questions and hopefully you could help me. To give you a bit of context I'm working in the Southern Ocean area between New Zealand and the Ross Sea with acoustic data. I've got several transects running almost continuously across the the Southern Ocean, between the two points described above. So my questions are: first, as I'm working on this area I'm looking for the best plots available and I've tried to find a definite answer to this but I have been able to do so. I have plotted one of my transects using package map and mapdata, using a stereographic projection (see figure attached) but there could be a better one, any suggestion about what projection/package i should use when working on this area? My second question is about how to plot bars along a track, which represent a certain property. For example in the second plot across the South Pacific Ocean I plotted acoustic backscatter – just a measure of abundance of fish every 10 nautical miles and the chance of colour just represent day and night along the transect. I made this plot on ArcGIS but I as want to do it on R, perhaps not exactly the same but similar concept. Do you know if there is a package that allows this plotting capacities? Thanks beforehand

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Hi Pablo, I'm glad the post was useful. Unfortunately I don't think I have too much help to offer on your questions. I haven't looked around for other data packages, so I don't't know if there is better data on your area of the globe. If you can find shapefiles for the area online, that will probably be the most useful to you. Secondly, I don't know of any packages meant to do what you describe with the bar plots, but you could set up a figure like you describe just by drawing a segment for each point of data you have at the specified location on the map, and of the height you desire based on your data. There are probably simpler ways of going about it though, so perhaps searching through some of the packages on CRAN will help.

    Good luck!
    Kim

- Pingback: Spatial and movement analyses in R: links | Marco Plebani

- Pingback: How to create maps in R | funature blog

- *Jake Beam*
  Kim, this post is amazing! I made a map with my sample sites in < 1 day, thanks so much for taking the time to post this.

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Excellent, glad to hear it worked so well for you!

- *DP*
  Hi Kim,

  I will like to thank you for these handy R tools.

  Maybe someone in the molecular ecology community can help me!

  I am trying to plot Belarus but with the library maps, it is no possible!!!

  Have you an idea how to deal with this problem? By worldHires… is not Belarus!!!

  I was trying with the mapsfile from http://gadm.org/, but now the problem I cannot plot two countries in one plot!

  library(sp)

  library(maptools)

library(rgeos)

BLR<- readShapePoly("..................Belarus/BLR_adm1.shp")

GE<- readShapePoly ("....................Germany/DEU_adm1.shp")

PO<- readShapePoly("................Poland/POL_adm1.shp")

par(mar = (0,0,0,0))

plot(PO, col = "white", border = "black", asp=1.5,add=TRUE)

Please can you help me to resolve this problem

I would like to thank you in advance!!

Diego

- *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
  Hi Diego, I received your email, so I will take a look. At first glance I am not sure, so if anyone has encountered this before, feel free to chime in!

- Pingback: daily 02/05/2014 | Cshonea's Blog

- *TechUser2011*
  Thanks for the great article. How did you generate the detailed map used in the image showing Giles county:

  http://www.molecularecologist.com/wp-content/uploads/2012/09/MetapopBySizeFull-1024×666.png

  Are those fine lines in the map representing streets or rivers? Do you know of an R map dataset that has streets or highways?

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    So the thin lines are representing roads and the thickest ones are county lines, and those are each from a different shapefile for the data. They're read into R with "readShapeLines()", similar to readShapePoly for my example with the species range plotted.

    As for getting shapefile data, there might be road data in R out there somewhere. I know there is at least city data and county data in the mapdata package I think, so roads might exist. I got this data from the VADOT website if I remember correctly, so I'd recommend just googling around for the area you want or searching within CRAN. It is also possible to create your own shapefiles, but I do not know anything about that besides its existence.

- *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
  Hi all, I should also add this here. There is a recently released package for making interactive maps in R which looks to be quite fantastic and useful! http://rmaps.github.io/

- *Tanja Pyhäjärvi*
  Hi, great post! Really useful and well written. Someone already pointed out that Belarus is missing, but also other countries are missing from worldHires too. Where is Estonia, for example? Any suggestions for better basemaps? This map is from The CIA World Data Bank II, and apparently not updated since 1980′s 😉

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Hi Tanja, yes good question, one that has come up a lot. Since writing this post, I've learned of another useful R package, "rworldmap' and it has more updated mapping data, including in Europe. You can find its description here: http://cran.r-project.org/web/packages/rworldmap/index.html

- *Dave Ensing*
  Thanks for this Kim; I find myself returning to this post time and again.

I haven't yet tried it, but perhaps the species distribution modellingpackage 'SDMTools' appears to have a solution for the projected map / scale bar issue you mentioned (http://www.rforge.net/doc/packages/SDMTools/Scalebar.html).

Also for people looking to use ggplot2 and to do GIS analysis in addition to making maps this relatively new tutorial (and associated website) seems pretty awesome: http://spatial.ly/2013/12/introduction-spatial-data-ggplot2/

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Hi Dave, thanks for this! I'll definitely have to check out SDMTools. There is also a ggmap package out there which is probably great; I'm just personally not a ggplot user. Thanks again for the links!

- *Meraj Duaa*
  Hi!!! I am the student of M.Phil (Statistics) can you help me about disease mapping??? I am too much confused about my work of disease mapping in my district.

  Thanks Kim.

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Hi Meraj, if you have a specific question about this post or on mapping with R, you can ask it here.

- Pingback: R365: Day 31 – RgoogleMaps | Something Like Science

- Pingback: Mas eu ainda quero meu hamburger | De Gustibus Non Est Disputandum

- *Howard*
  Thanks a lot for sharing! I found this post extremely useful!

- *http://www.unr.edu/~cmmoore Christopher Moore*
  Thank you. This is a pretty darn good resource. I may borrow some of this to share (and of course give you credit). It's just really straightforward and easy. Thanks!!!

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Hi Christopher, great, glad to hear you found it helpful! Have fun mapping!

- Pingback: Projecting a map to match your data in R | On the other hand

- *Dan*
  This is incredible. We are researchers at the New York State Department of Health, working on a AIDS mapping project and this is key. Really appreciate your work on this.

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Thanks Dan, glad to hear it is getting put to good use!

- *Alexandre*
  Kim, Could you show the #to plot all points, I run a loop to run through my data one point at a time and make each pie chart; there are most likely more efficient methods example?
  Thanks,

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Hi Alexandre, eershova actually had a good comment about this above, that instead of a loop, one could do:

    "Rather than running a loop, to plot multiple pies you can use the draw.pie command. Super easy to use once you transform your data into the right format using "make.xyz"."

    I think that is probably much better than what I did, though I haven't tried it. I haven't made such a map since this post, but looking back at my old, inefficient code, this is how I did it:

    ids <- seq(1, length(gps$POP)) #make a list the length of my number of data points

```
i <- 0 #start the loop at 0
for(i in ids){ #run through all data points
east <- gps[i,2] # the blue parts of the pies
west <- gps[i,3] # the orange parts of the pies
tot <- gps[i,4] # the size of the pie (radius)
lat <- gps[i,5] # latitude
lon <- gps[i,6] # longitude
add.pie(z=c(east, west), x=lon, y=lat, radius=sqrt(tot)/2, col=c(alpha("orange", 0.6), alpha("blue", 0.6)), labels="")
}
```

- *Confused*
  Great tutorial thank you- but I am stuck! I just want to plot a simple map with sampling points and I get the map but no points.

  map("worldHires","UK",xlim=c(-8.26,2),ylim=c(49,59.5))
  points(data$Lat,data$Long,pch=19,col="green",cex=0.5))

  What am I doing wrong? I have checked the points are definately within the map range.

  Thanks

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Without seeing the rest of your code and the data, I am not sure what the error is. Feel free to email me; follow the link in the post to my site for contact info.

- *Peter*
  Hi Kim, this is very useful and highly informative. I was wondering if you know how to add a third dimension e.g. depth. I work with marine animals and would like to map marine fish to depth. Do you know if this is possible?

  - *http://www.zoology.ubc.ca/~kgilbert/mysite/Welcome.html kimgilbert*
    Hi Peter, unfortunately I don't know how to do that offhand. I know that 3-D graphics do exist in R but have never delved into them. You could indicate depth by contours or shading like a heat map if you have the data for that, but I'd say try searching around a bit and maybe there are some packages to help you out. Good luck!

- *Jack*
  Kim, I was not able to get past the easy step. This is my code and the error message that I receive:

  #Downloading Packages
  #install.packages("maps")
  #install.packages("mapdata")

  #Loading packages
  library("maps")
  library("mapdata")

  #Basic Test
  map("worldHires","Canada", xlim=c(-141,-53), ylim=c(40,85), col="gray90", fill=TRUE)

  Error: unexpected symbol in "map("worldHires","Canada", xlim=c(-141,-53), ylim=c(40,85), col="gray90″

  Any pointers in the right direction would be appreciated.
  Thank you!
  ~J