How to make a GEOmap

Jonathan M. Lees
University of North Carolina, Chapel Hill
Department of Geological Sciences

January 29, 2014

1 Introduction

A GEOmap is a set of lists that describes a geographic map with meta data information that makes it easy to plot complicated structures, like geological maps.

If you have simply a set of LAT-LON pairs you want to plot, this can be very easy.

You start by reading in the data. I made a a small map outline by clicking on a map, but you can get maps from many different sources.

Here I dump the clicks to a file:

```
> library(GEOmap)
>
> ##  write.table(L, file = "testmap.txt", append = FALSE, sep = " ",
> ##                  eol = "\n", na = "NA", dec = ".", row.names = FALSE,
> ##                   col.names = FALSE)
```

Next I read in the data, as if it were stored in a file:

```
> ##    latlon = scan(file='testmap.txt', what=list(lon=0, lat=0) )
>
```

```
> latlon=list()
> latlon$lat=c(39.8780395624,39.7488080389,39.4903449921,39.2964977069,
 39.1995740643,39.1349583026,38.9088031365,38.6180322088,38.3272612810,
 38.0041824724,37.8749509489,37.8749509489,38.3272612810,38.4888006853,
 38.8118794939,39.0057267791,39.2318819452,39.5872686346,39.9426553241)
> latlon$lon=c(136.6629878969,136.3444990720,136.0715086507,136.0715086507,
 135.6165246151,135.0250453689,134.9795469653,134.9795469653,
 135.0705437724,135.2525373866,135.7530198258,137.0724735289,
 137.3454639502,137.4364607574,138.0734384071,138.0734384071,
 137.8004479858,137.7549495822,137.2544671431)
>
>
>
```

Plot the raw data as a check:

```
> postscript(file="./fig/H2map1.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plot(latlon$lon, latlon$lat, pch=1)
> lines(latlon$lon, latlon$lat)
> dev.off()
>
```

The data look okay, prepare a GEOmap:

```
> mymap  = list(STROKES=list(nam="map1", num=length(latlon$lat), index=0, col=rgb(1,.7, 1)
 POINTS=list(lat=latlon$lat, lon=latlon$lon))
> mymap = boundGEOmap(mymap)
>
```

Plot with no projection:

```
> plotGEOmap(mymap)
>
```

Plot with projection:

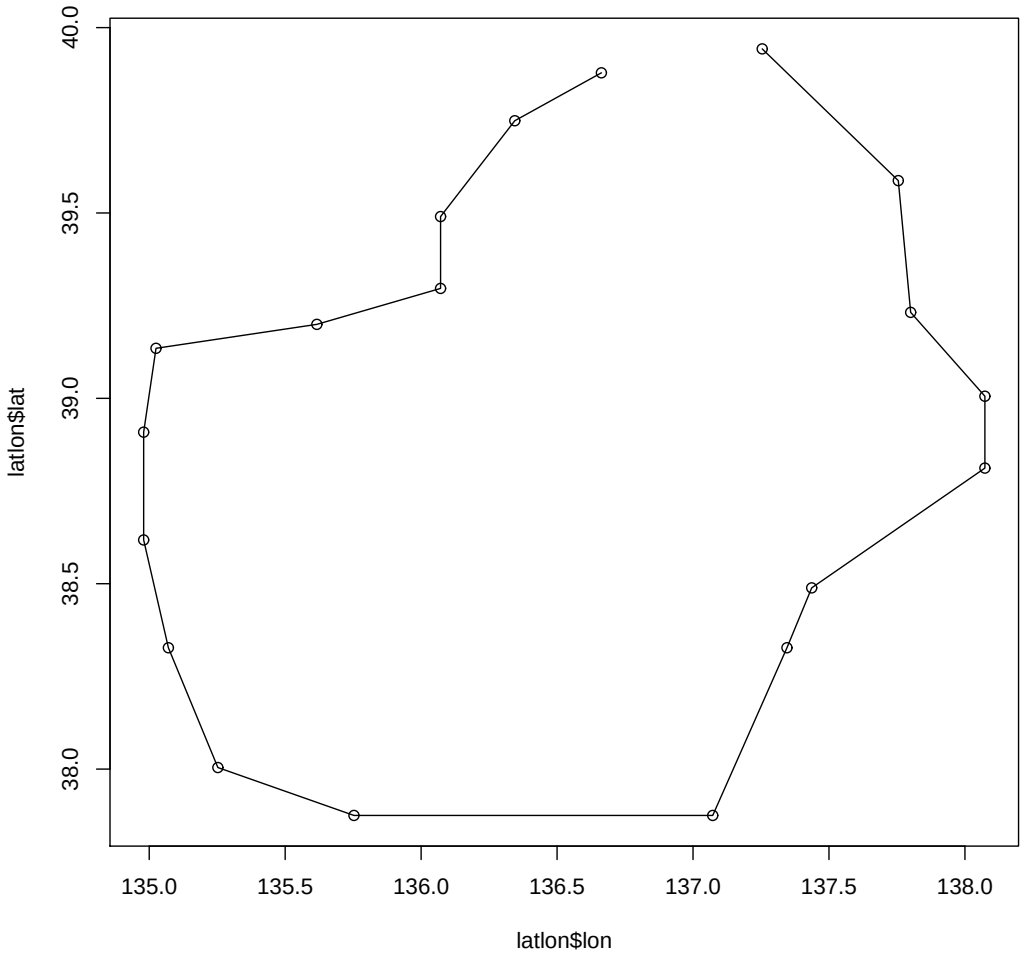

Figure 1: H2map1

```
> proj = setPROJ(type=2, LAT0 = mean(latlon$lat), LON0 = mean(latlon$lon))
> postscript(file="./fig/H2map2.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plotGEOmapXY(mymap, PROJ=proj)
> dev.off()
>
>
```

add outlines:

```
> postscript(file="./fig/H2map3.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
```

```
> plotGEOmapXY(mymap, PROJ=proj)
> plotGEOmapXY(mymap, PROJ=proj, MAPstyle=2, MAPcol='black', add=TRUE , linelwd=2 )
> plotGEOmapXY(mymap, PROJ=proj, MAPstyle=1, MAPcol='black', add=TRUE , ptpch=1 )
> dev.off()
>
```

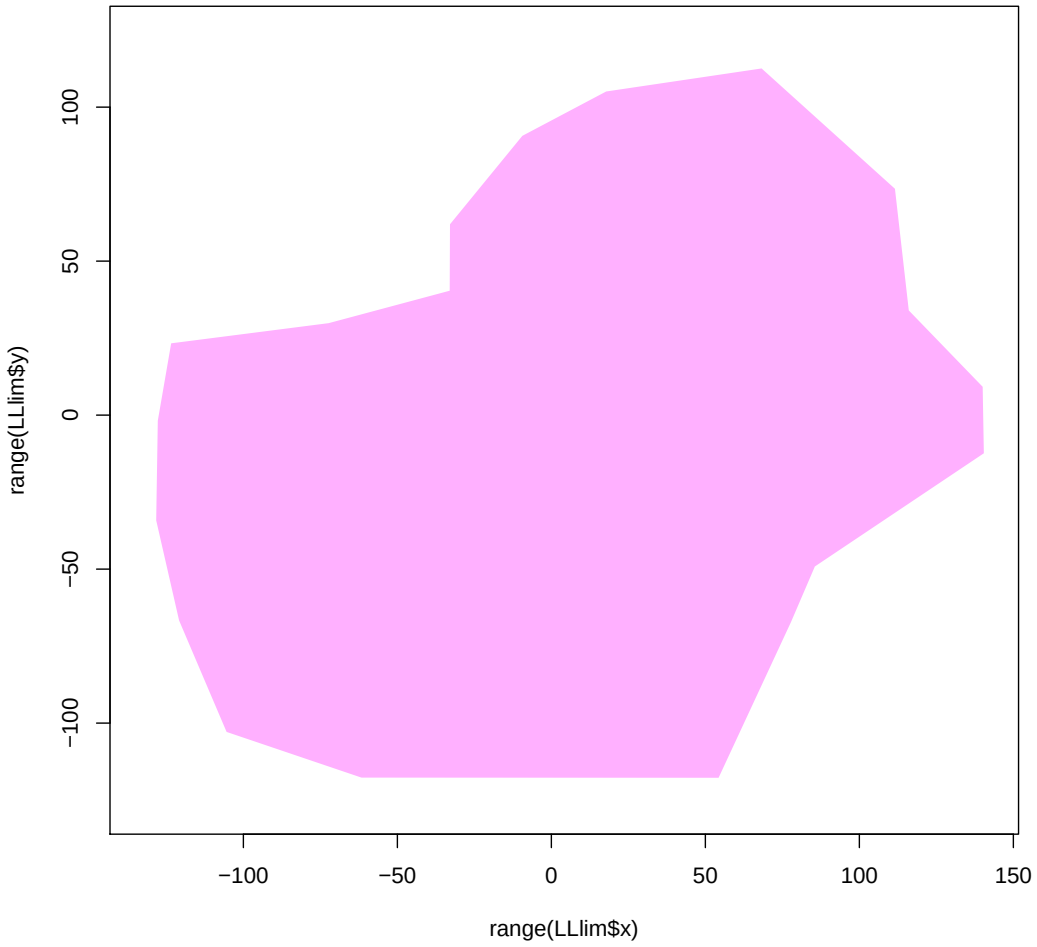

Figure 2: H2map2

Here we add information by simple projection and standard R commands:

```
> postscript(file="./fig/H2map4.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plotGEOmapXY(mymap, PROJ=proj)
> XY = GLOB.XY(latlon$lat, latlon$lon, proj)
> points(XY)
```

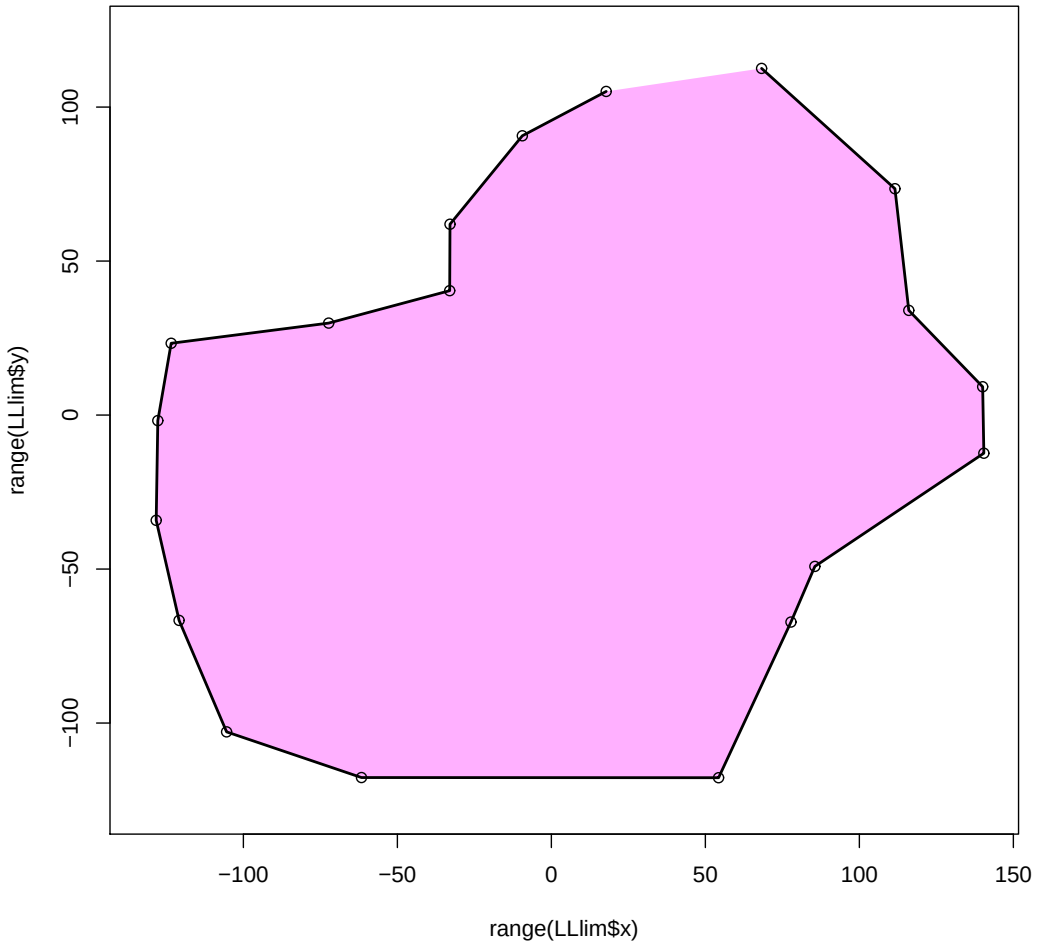

Figure 3: H2map3

```
> lines(XY)
> dev.off()
>
```

Finally, lets see this map added to a larger map

```
> library(RPMG)
> library(GEOmap)
> library(geomapdata)
> data(japmap)
> postscript(file="./fig/japmap1.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
```

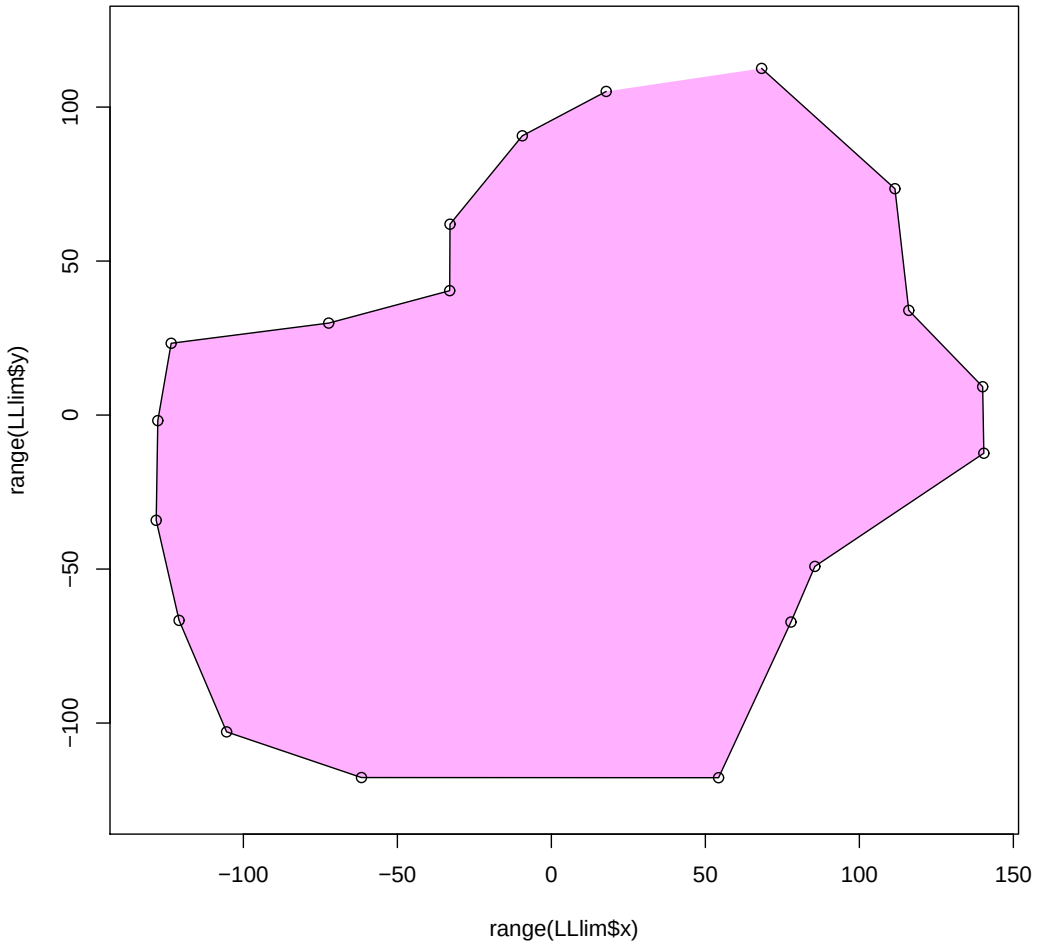

Figure 4: H2map4

```
> plotGEOmapXY(japmap, PROJ=proj)
> plotGEOmapXY(mymap, PROJ=proj, add=TRUE)
> dev.off()
>
>
```

2 Multiple strokes

The previous example showed a case where we had only one stroke in the map. Usually a map will consist of many strokes, each with possibly different attributes (style, color, etc.)

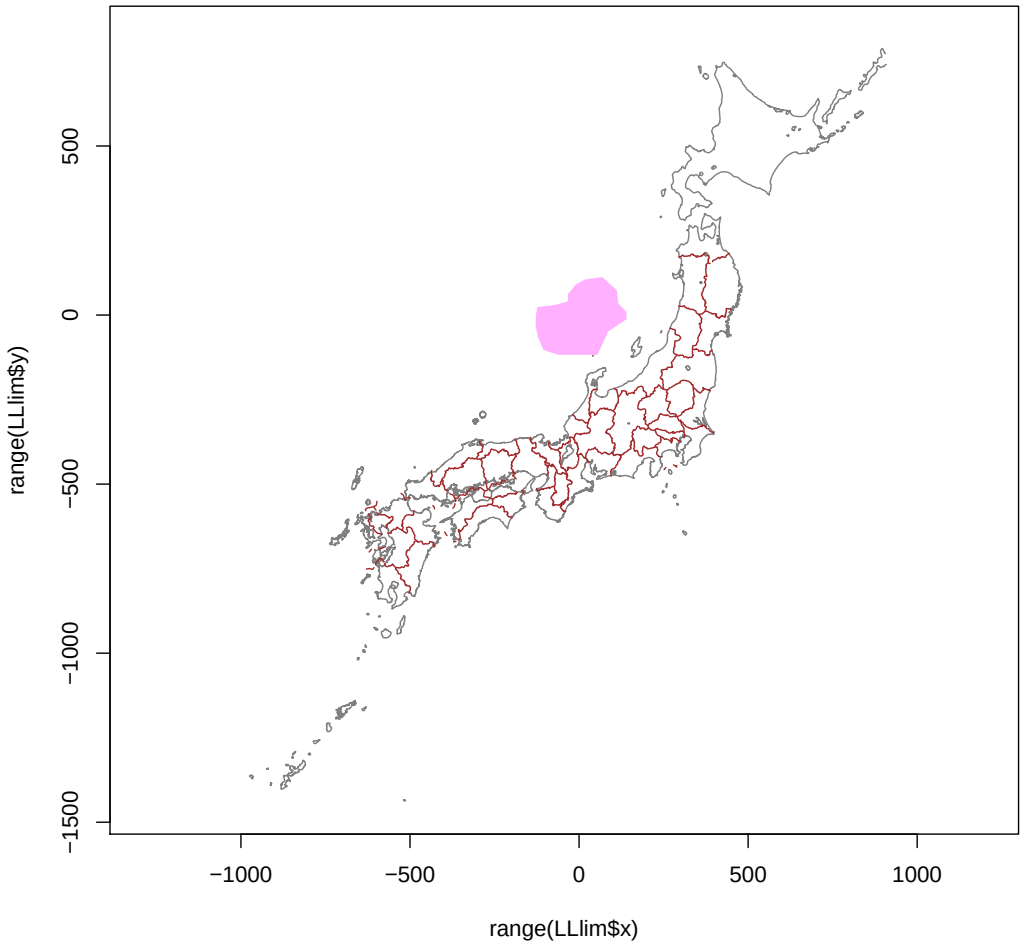

Figure 5: japmap1

Consider the following line:

```
> GLL=list()
> GLL$lat=c(38.0552647517,38.1533772893,38.2754431875,
 38.3672221979,38.5260793869,38.6483246519,38.7701056377,
 38.8976069603,38.9457673342,38.9998962787,39.1025327692,
 39.1927889270,39.3801557421,39.5193850467)
> GLL$lon=c(135.7446171004,135.8598134616,135.9053532164,
 135.9978522791,136.1369466401,136.3703056863,136.6044613488,
 136.8081531656,136.9649782331,137.1064020435,137.2564343909,
 137.4067379892,137.5747171917,137.6637851576)
> mymap2  = list(STROKES=list(nam=c("map1", "map2"),
 num=c(length(latlon$lat), length(GLL$lat)),
```

```
 index=c(0, length(latlon$lat)) ,
 col=c(rgb(1,.7, 1), "blue") ,
 style=c(3,2),
 code=c("o", "thrust") ),
 POINTS=list(lat=c(latlon$lat, GLL$lat), lon=c(latlon$lon, GLL$lon)) )
> mymap2 = boundGEOmap(mymap2)
> postscript(file="./fig/H2map5.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plotGEOmapXY(mymap2, PROJ=proj)
> dev.off()
>
>
```

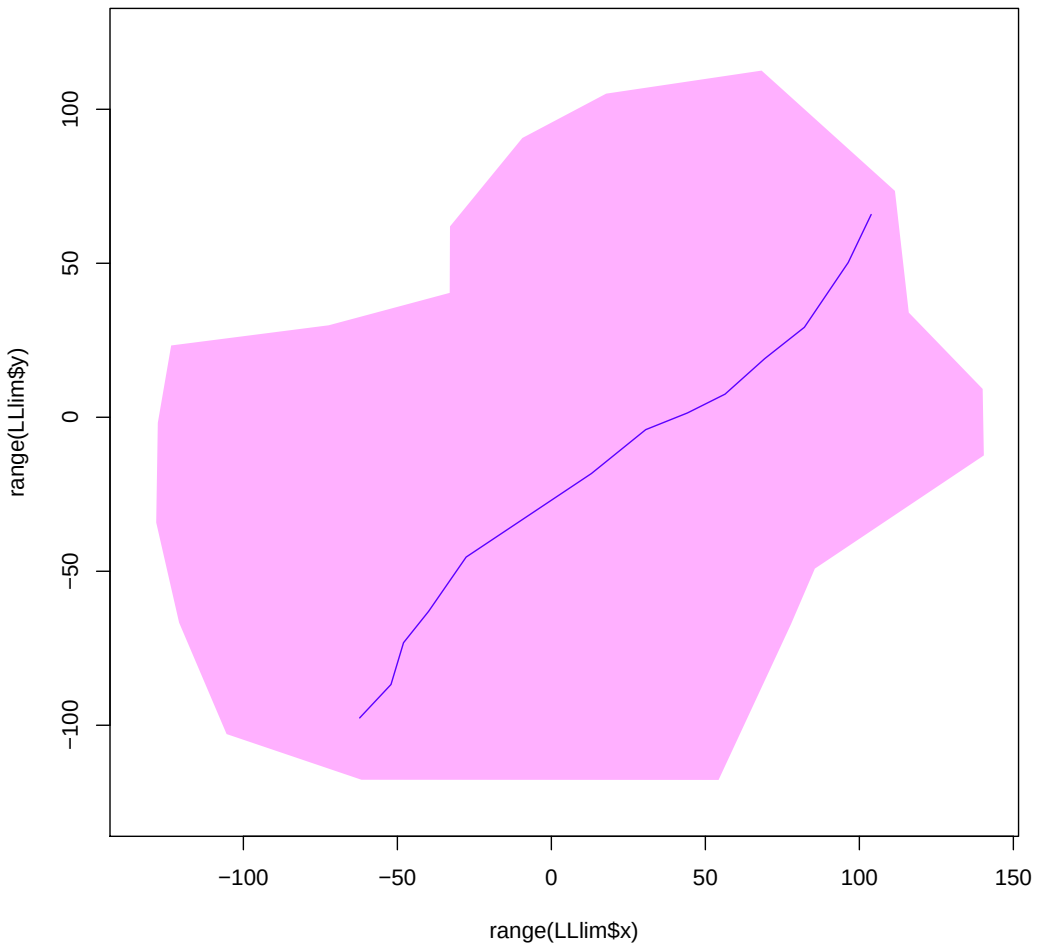

Figure 6: H2map5

Make a geologic map

```
> GXY = GLOB.XY(GLL$lat, GLL$lon, proj)
> postscript(file="./fig/H2map6.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plotGEOmapXY(mymap2, PROJ=proj)
> thrust(GXY$x, GXY$y,  h=4, N=20, REV=FALSE)
> dev.off()
>
>
>
```

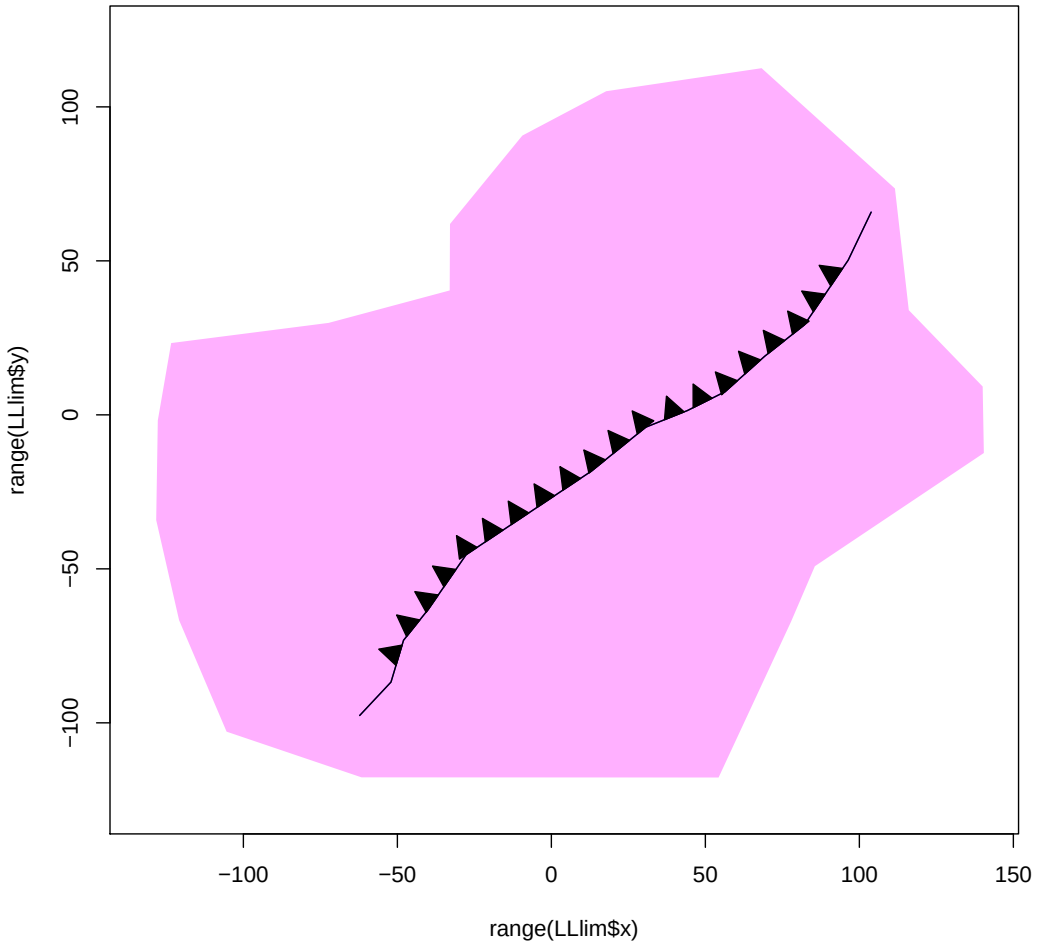

Figure 7: H2map6

3 Plotting with GEOmaps

Plotting and constructing figures with GEOmap can be made easy by using the plotting functions:

linesGEOmapXY add lines to a plot

rectGEOmapXY add rectangles to a plot

textGEOmapXY add text to a plot

pointsGEOmapXY add points to a plot

See the documentation for examples and illustration on how to use these. There is also a vignette with more documentation on GEOmap.

4 Manipulating GEOmaps

There are several utility functions for manipulating GEOmap list structures. These can be used to edit or modify existing GEOmap lists, as well as combine strokes or merge whole databases.

list.GEOmap Convert a list of strokes to a GEOmap

GEOmap.cat Dump out a GEOmap

GEOmap.Extract Extract a set of strokes from a GEOmap

GEOmap.limit Extract a set of strokes from a GEOmap, based on Lat-lon limits

GEOmap.CombineStrokes Combine strokes in a GEOmap

ExcludeGEOmap exclude strokes from a GEOmap

SELGEOmap Select strokes from a GEOmap

maps2GEOmap Convert data from the maps package to GEOmap

4.1 Map from list

By way of illustration, another way to create a GEOmap is to first create a list of strokes and then convert them using the utility *list.GEOmap*

```
> LL = list()
> LL[[1]] = list(lat=latlon$lat, lon = latlon$lon)
> LL[[2]] = list(lat=GLL$lat, lon = GLL$lon)
> J = list(LL=LL)
> GL = list.GEOmap(J)
```

Plot the result, as before, with a default projection:

```
> postscript(file="./fig/H2map7.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plotGEOmapXY(GL)
> dev.off()
>
>
```

4.2 Map Limited by Lat-Lon

It may be inconvenient to us a very large map (many strokes) when you are working on a much smaller geographic region. If you are plotting and replotting a map in avariety of different contexts and you expect to work in this region a lot with the same mapping parameters it is faster to extract fromt he larger map data base a smaller subset of the massive map data set.

```
> library(GEOmap)
> staf = "signetsta.LLZ"
> sta = scan(file=staf, list(name="", lat=0, lon=0, z=0))
> attr(sta, "filename") = staf
>  PLOC=list(LON=c(-105, -77.50000),LAT=c(-5, 11),
                  x=c(-105, -77.50000), y=c(-5, 11) )
> PROJ = setPROJ(type = 2, LAT0 = median(sta$lat) , LON0 = median(sta$lon))
> ils  = list.files(path=".", pattern="RDATA" )
> ig = grep("sgalap.RDATA", ils)
```

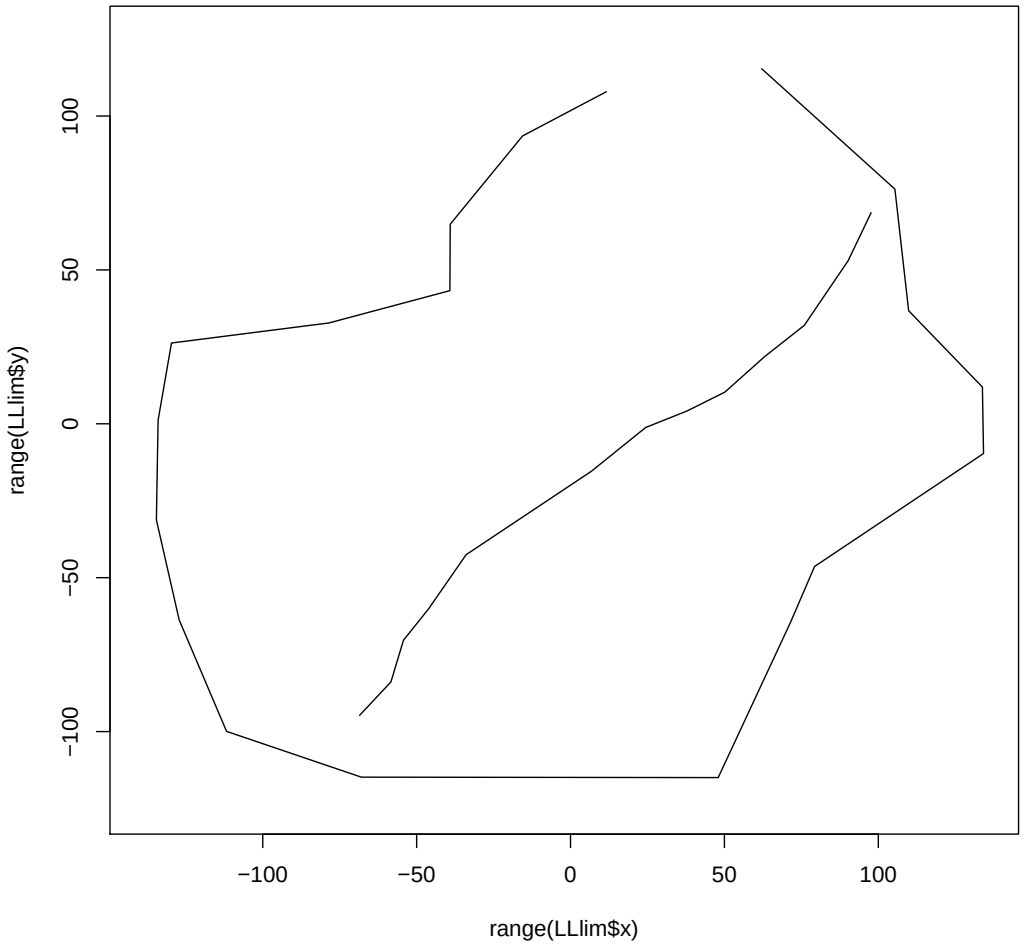

Figure 8: H2map7

```
> if(length(ig)<1)
   {

 library(geomapdata)


 library(WORLDMAP2)

 data(world2)


 data(coastmap)
 data(world2)
```

```
data(worldmap)

sgalap  = SELGEOmap(world2, ncut=3, acut=c(10, Inf), proj=PROJ,
  LIM=c(PLOC$LON[1], PLOC$LAT[1],PLOC$LON[2], PLOC$LAT[2] ))

save(file="sgalap.RDATA", sgalap)

}else{
   load("sgalap.RDATA")
  }
```

Temporarily plot the whole map with now projection:

```
> plotGEOmap(sgalap)
> points( fmod(sta$lon, 360) , sta$lat, pch=6, col="red", cex=.6 )
```

Note that the figure and map is too large to be of use for detailed analysis of the Galapagos. This map might be used for showing ocean topography. There are some parts missing off the west coast of South America, however.

Next we use this large map file to plot a more detailed area:

Start by setting up the bounds for the map. Bounds for the stations are found and then expanded slightly for a nicer plot. For plotting limits the convention is x1,y1,x2,y2 (or Lon1,Lat1,Lon2,Lat2.

```
> maplim1 = c(min(sta$lon), min(sta$lat), max(sta$lon), max(sta$lat)   )
> LONS = expandbound(range(sta$lon), .15)
> LATS = expandbound(range(sta$lat), .15)
> maplim1 = c(LONS[1], LATS[1], LONS[2], LATS[2])
```

```
> postscript(file="./fig/G2map1.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plotGEOmapXY(sgalap, LIM=maplim1 ,PROJ=PROJ, add=FALSE, xlab="km", ylab="km"  )
> pointsGEOmapXY(sta$lat, sta$lon,PROJ=PROJ, pch=6, cex=.6, col='red')
> textGEOmapXY(sta$lat, sta$lon,labels = sta$name , PROJ=PROJ, pos=3, cex=.6)
> dev.off()
```

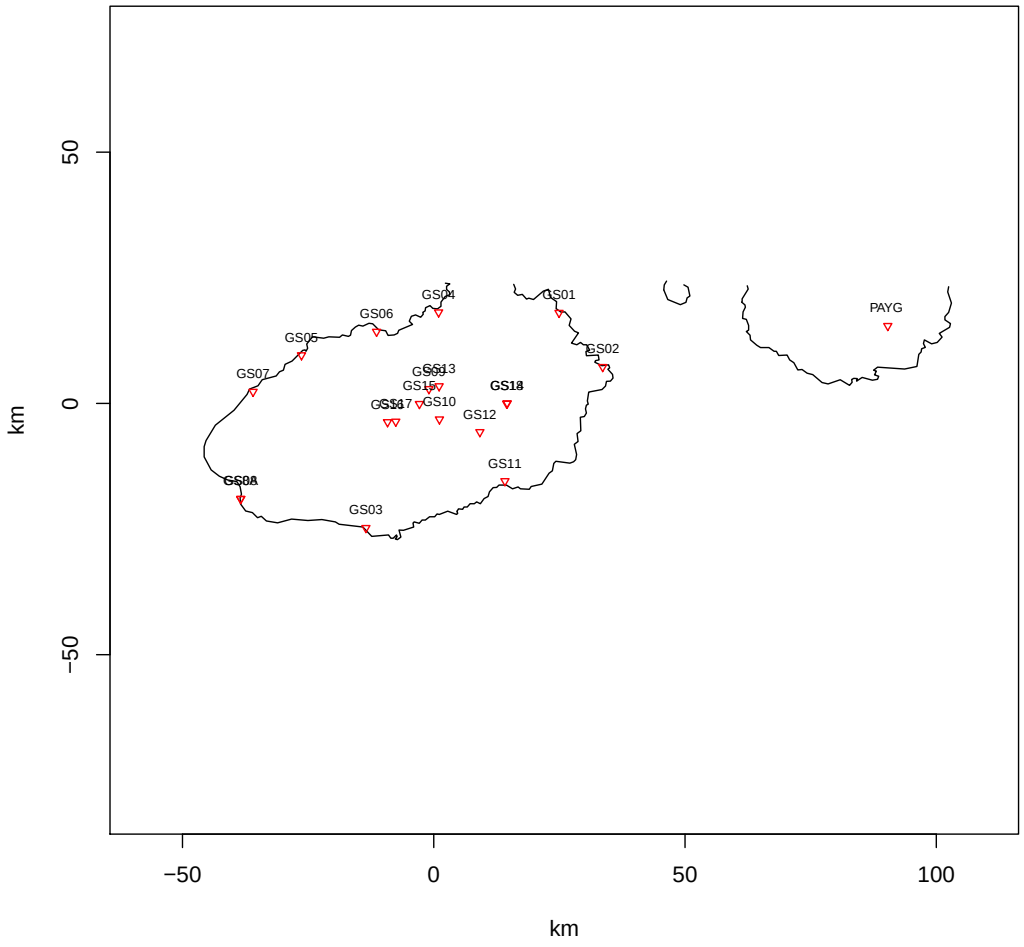

Figure 9: G2map1

This plot looks okay, but really we will want to use the map many times so we should restrict the map database to just the Galapagos Archipelago, at least a first.

First replot the data, as in the previous unprojected plot. Then select a rectangle for extraction: use the locator() function. After extraction we plot the mapDB with no projection to check it out.

```
> #####   J = locator(2)
> J=list()
> J$x=c(267.864124391,271.486129954)
> J$y=c(-1.98038370386, 1.06604025528)
> LLX = list(lon=J$x, lat=J$y)
> GALmap = GEOmap.limit(sgalap, LLX )
> postscript(file="./fig/G2map2.eps", width = 8, height = 8, paper = "special",
```

```
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
>  plotGEOmap(GALmap)
> points( fmod(sta$lon, 360) , sta$lat, pch=6, col="red", cex=.6 )
> dev.off()
```

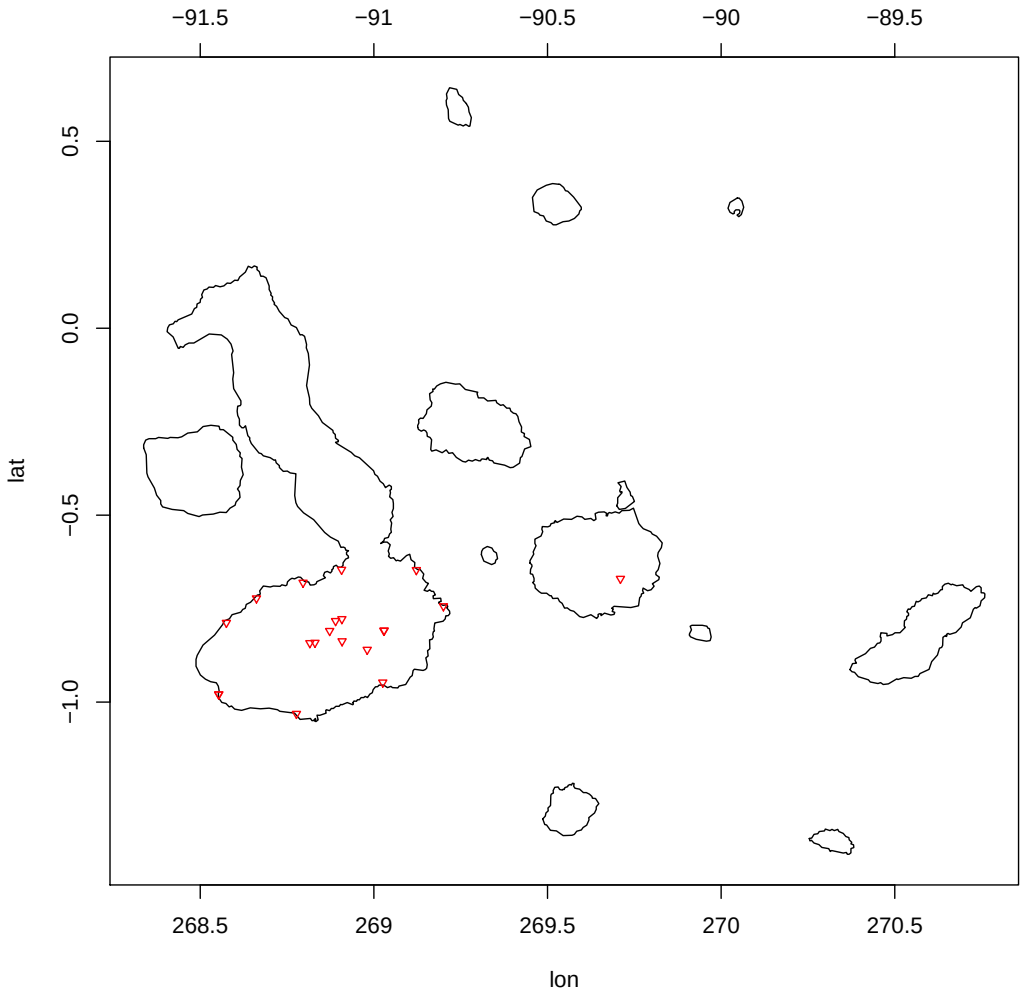

Figure 10: G2map2

Note that the western coast of South America is now removed from the database.

If we wish to get just the largest of the Galapagos Islands as a mapDB, it would be hard to extract with a Lat-Lon target since the island Fernandina is surrounded by Isabella. In this case we can extract by selecting the number in the database.

First, plot the data with the stroke numbers shown:

```
> postscript(file="./fig/G2map3.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
>  plotGEOmap(GALmap, NUMB=TRUE)
> dev.off()
```

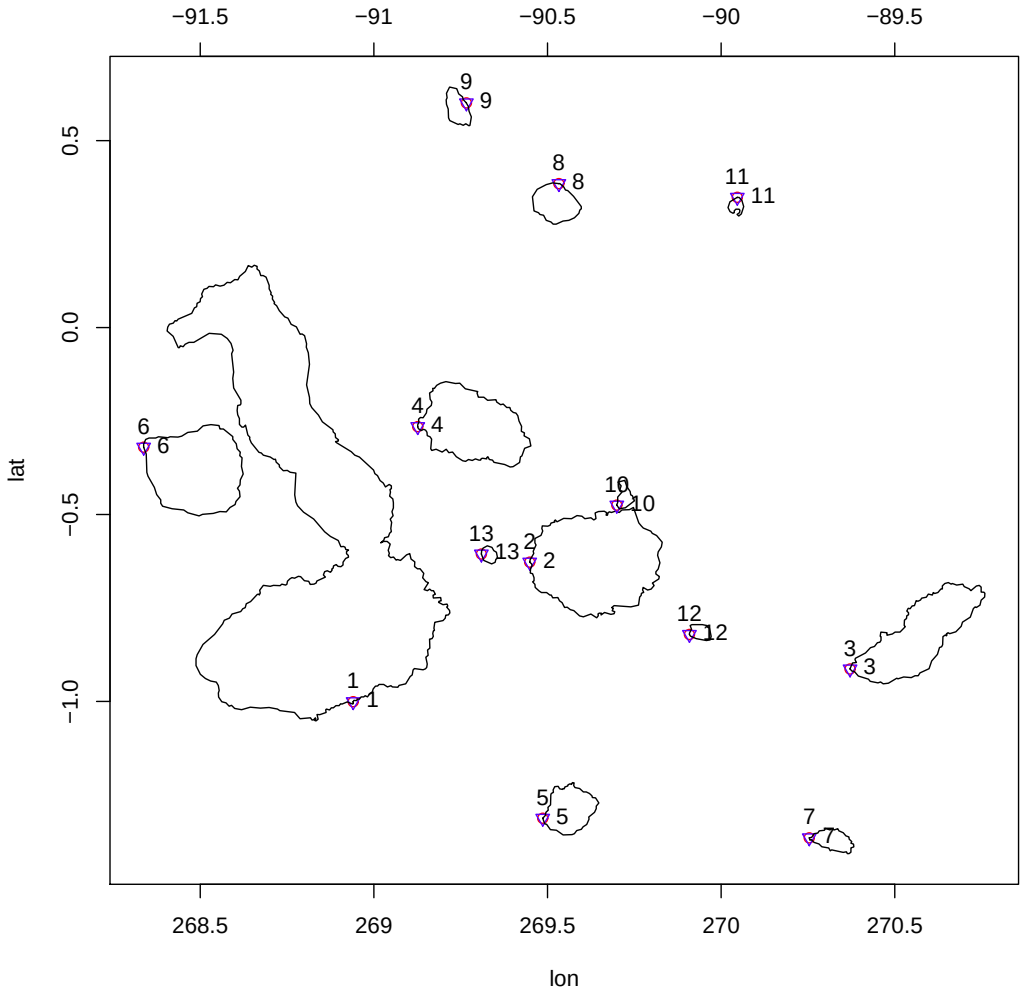

Figure 11: G2map3

and see that the first stroke in the database is all of Isabella. The sixth stroke is Fernandina.

```
> ISAB = GEOmap.Extract(GALmap,1, INOUT="in" )
> FERN = GEOmap.Extract(GALmap,6, INOUT="in" )
> postscript(file="./fig/G2map4.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> par(mfrow=c(1,2))
>  plotGEOmap(ISAB)
```

```
>  plotGEOmap(FERN)
> dev.off()
```

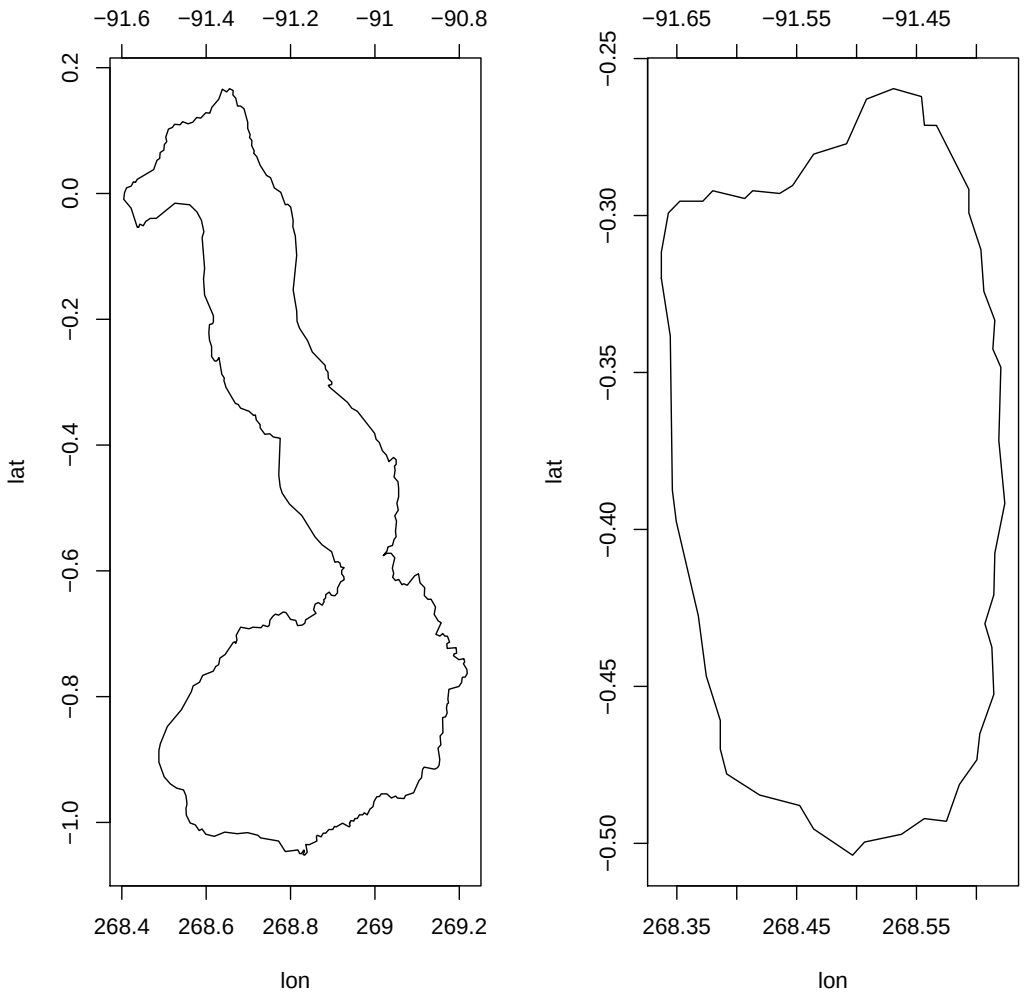

Figure 12: G2map4

We can adjust the database here to make sure these are filled polygons:

```
> ISAB$STROKES$style = 3
> FERN$STROKES$style = 3
> ISAB$STROKES$col = rgb(.8, .8, 1)
> FERN$STROKES$col  = rgb(1,.8,.8)
> postscript(file="./fig/G2map5.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> par(mfrow=c(1,2))
```

```
>  plotGEOmap(ISAB)
>  plotGEOmap(FERN)
> dev.off()
```

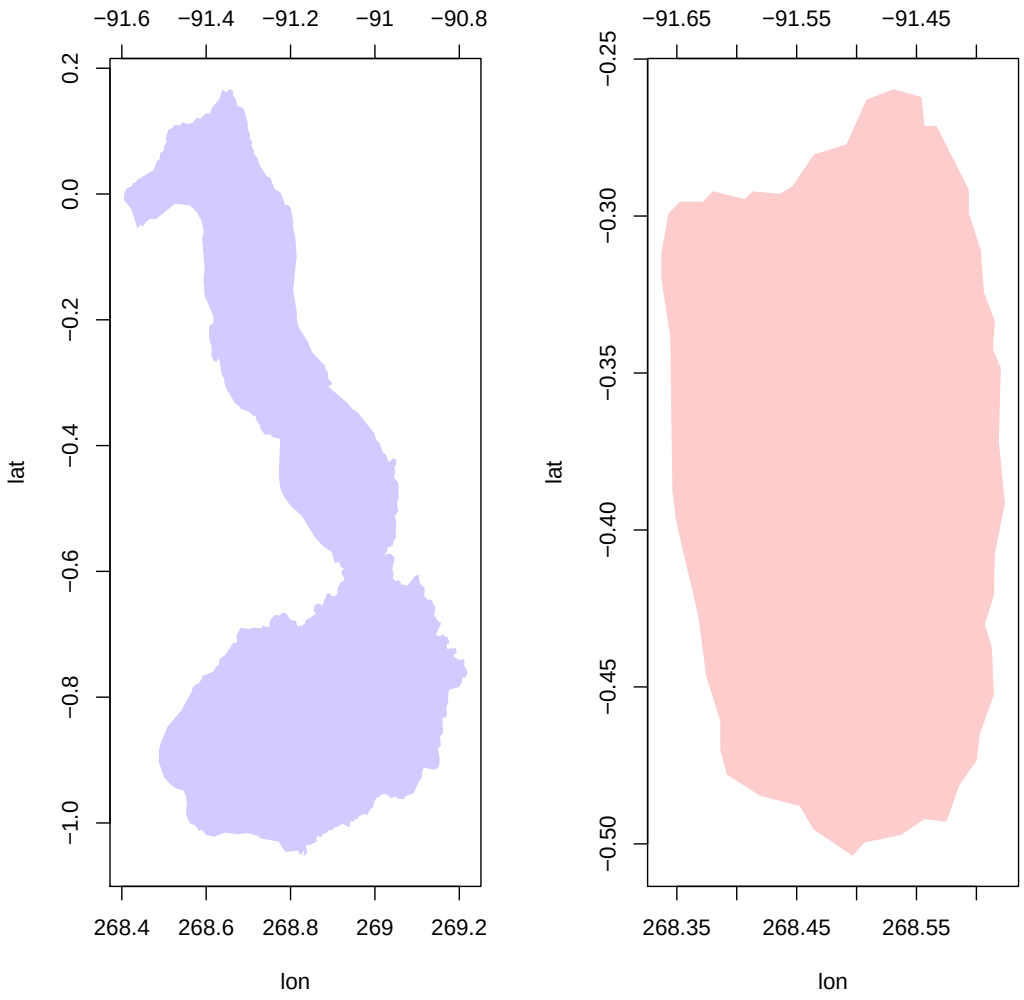

Figure 13: G2map5

We are still focussed on teh southern end of Isabella where the voclanoes, Sierra Negra and Cerro Azul reside. So next we can cut down the ISAB map to show only the southern extent.

```
> ISAB$STROKES$style = 2
> m2 = c(fmod(-91.62032 , 360), -1.08901 , fmod(-90.11648  , 360), -0.58799)
> postscript(file="./fig/G2map6.eps", width = 8, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plotGEOmapXY(ISAB, LIM=m2 ,PROJ=PROJ, add=FALSE, xlab="km", ylab="km"  )
```

```
> pointsGEOmapXY(sta$lat, sta$lon,PROJ=PROJ, pch=6, cex=.6, col='red')
> textGEOmapXY(sta$lat, sta$lon,labels = sta$name , PROJ=PROJ, pos=3, cex=.6)
> dev.off()
>
```

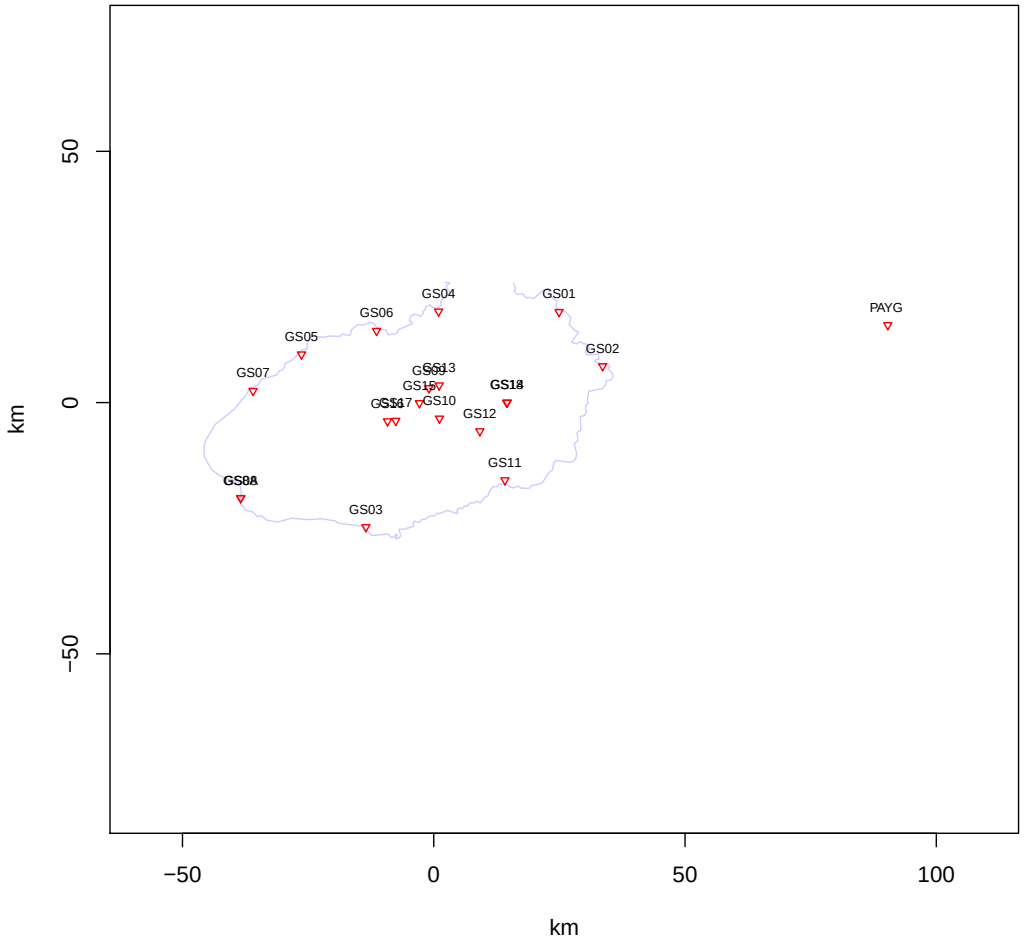

Figure 14: G2map6

Here we limit the map one last time to extract the southern end of Isabella Island.

```
> H=list()
> H$x=c(268.43,269.28)
> H$y=c(-1.09107,-0.57066)
> LLX = list(lon=H$x, lat=H$y)
> GALmap = GEOmap.limit(sgalap, LLX )
```

```
>  plotGEOmap(GALmap)
> points( fmod(sta$lon, 360) , sta$lat, pch=6, col="red", cex=.6 )
>
```

To add to the figure lines that are not part of the map database, first read them in:

```
> load("SNFAULTS.RDATA")
>
```

Finally plot them with the map:

```
> postscript(file="./fig/Gfin.eps", width = 10, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plotGEOmapXY(GALmap ,PROJ=PROJ, add=FALSE, xlab="km", ylab="km"  )
> for(i in 1:length(SNFAULTS))
   {
 propFAULT = SNFAULTS[[i]]
 linesGEOmapXY(propFAULT$lat, propFAULT$lon,PROJ=PROJ, col=grey(.6) )

   }
> pointsGEOmapXY(sta$lat, sta$lon,PROJ=PROJ, pch=6, cex=.6, col='red')
> textGEOmapXY(sta$lat, sta$lon,labels = sta$name , PROJ=PROJ, pos=3, cex=.6)
> dev.off()
```

5 *maps* Database: States

There are maps in R that are seperate from the GEOmap package.

The first part is used here to extract the data from the maps database.

To get a list of all the maps available for the United States, try this:

```
> library(maps)
> zednames  = map(database = "state", regions = ".", plot=FALSE, namesonly =TRUE  )
>
```

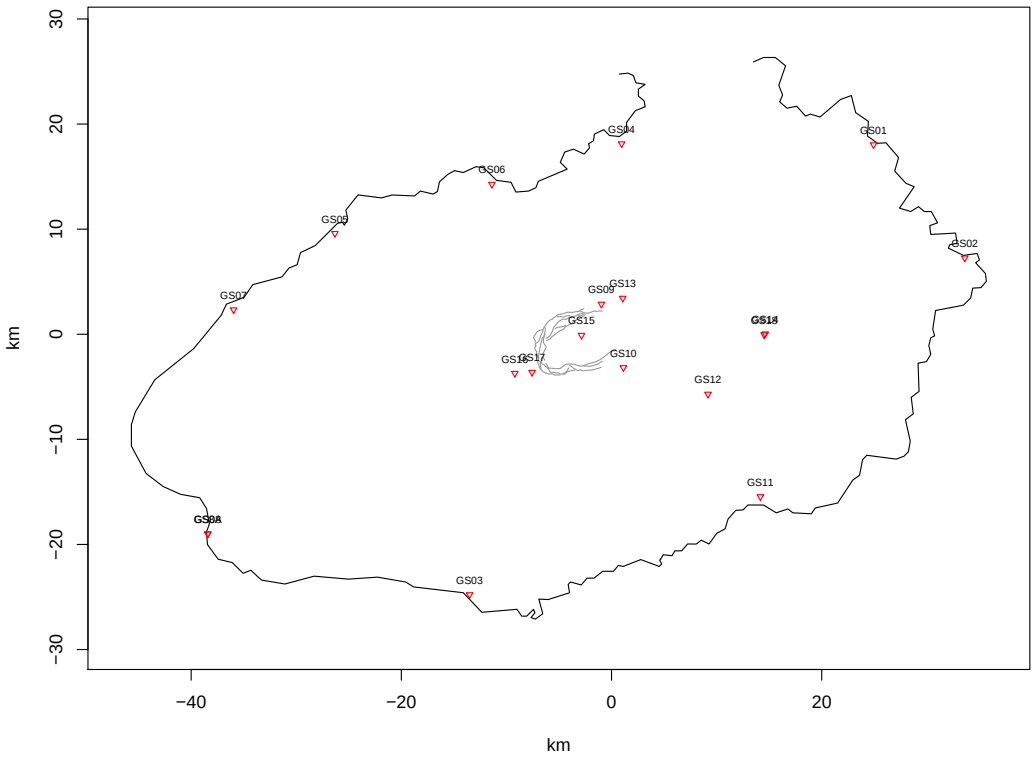

Figure 15: Gfin

To get a region, that is more selective, and the other states that are near by, try this:

```
> library(GEOmap)
> library(maps)
>  zz2 = map('state', region = c('north carolina', 'south carolina' , 'virginia' ),  plot = FA
> xlim = range(zz2$x, na.rm=TRUE)
> ylim = range(zz2$y, na.rm=TRUE)
>  zz2 = map('state', region = '.', xlim=xlim, ylim=ylim ,  plot = FALSE , fill=TRUE, namesc
>
```

The zz2 list has all the information in it, just need to reorganize to get the maps redone.

```
> SEmap  = maps2GEOmap(zz2)
> ###   get the bounds:
> SEmap  = boundGEOmap(SEmap)
> ###  expand the region:
> NClatlim = range(SEmap$POINTS$lat)
```

```
> NClonlim = range(SEmap$POINTS$lon)
> Lon1 = expandbound(NClonlim, pct = 0.05 )
> Lon1 = c(floor(Lon1[1]),  ceiling(Lon1[2] ) )
>  Lat1 = expandbound(NClatlim, pct = 0.05  )
> Lat1 = c(floor(Lat1[1]),  ceiling(Lat1[2] ) )
>   NClatlim =Lat1
>   NClonlim = Lon1
> ###   projection is utm (clark) with the center specified
> ncproj = setPROJ(type=2, , LAT0 =mean(NClatlim), LON0 = mean(NClonlim) )
> #### set up the boundary
> PLOC=list(LON=NClonlim , LAT=NClatlim)
> gxy = GLOB.XY(PLOC$LAT, PLOC$LON, ncproj)
>       PLAT =  pretty(PLOC$LAT)
>          PLAT = c(min(PLOC$LAT),
      PLAT[PLAT>min(PLOC$LAT) & PLAT<max(PLOC$LAT)],max(PLOC$LAT))
>        PLON  = pretty(PLOC$LON)
>              PLON = c(min(PLOC$LON),
      PLON[PLON>min(PLOC$LON) & PLON<max(PLOC$LON)], max(PLOC$LON))
>
```

```
> ###################  plotting here:
>
> postscript(file="./fig/TESTmaps.eps", width = 10, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plot(gxy$x, gxy$y,  asp=TRUE, ann=FALSE , axes=FALSE, type="n" )
>      plotGEOmapXY(SEmap, LIM=c(PLOC$LON[1], PLOC$LAT[1],PLOC$LON[2],
          PLOC$LAT[2]) , PROJ=ncproj, add=TRUE, SEL=1:length(SEmap$STROKES$num)   )
>      addLLXY(PLAT,  PLON, PROJ=ncproj , LABS=TRUE, PMAT=NULL, TICS=c(.1,.1) )
> dev.off()
>
>
```

Next, colorize the map:

```
> N = length(SEmap$STROKES$num)
> kcol = rainbow(N)
> SEmap$STROKES$col = kcol
> postscript(file="./fig/TESTmaps2.eps", width = 10, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> plot(gxy$x, gxy$y,  asp=TRUE, ann=FALSE , axes=FALSE, type="n" )
>      plotGEOmapXY(SEmap, LIM=c(PLOC$LON[1], PLOC$LAT[1],PLOC$LON[2],
```

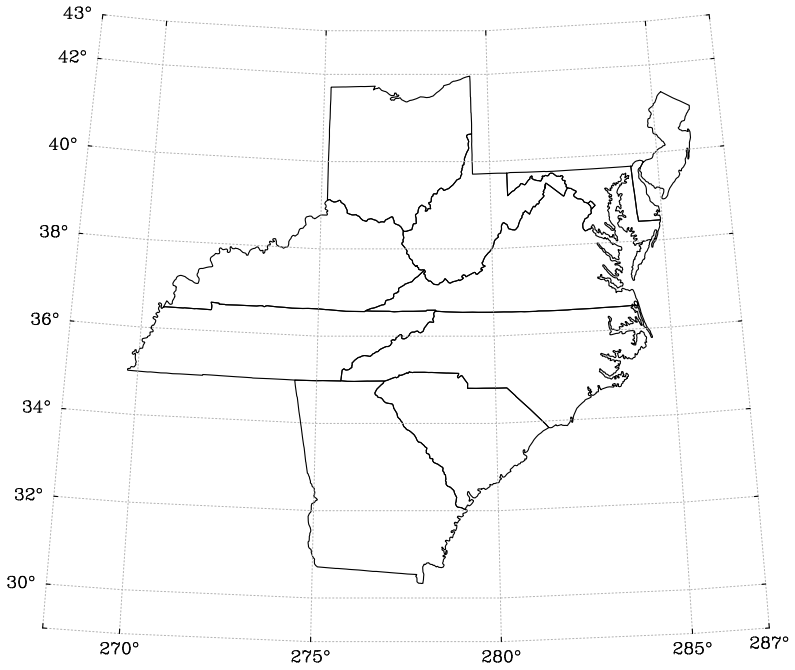

Figure 16: Gfin

```
          PLOC$LAT[2]) , PROJ=ncproj,MAPstyle=3, add=TRUE, SEL=1:length(SEmap$STROKES$num
>      addLLXY(PLAT,  PLON, PROJ=ncproj , LABS=TRUE, PMAT=NULL, TICS=c(.1,.1) )
> dev.off()
>
>



> library(maps)
> source("./CODE/BaddLLXY.R")
> source("./CODE/quickUScol.R")
> postscript(file="./FIGS/UTM_US_maps3.eps", width = 14, height = 8, paper = "special",
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> quickUScol()
> dev.off()
> postscript(file="./FIGS/UTM_US_TESTmapsBW3.eps", width = 14, height = 8, paper = "specia
        horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> proj = quickUScol(grey((1:100)/100)[46:95], seed=34)
> data(us.cities)
>         wcaps =which( us.cities$capital==2  )
```

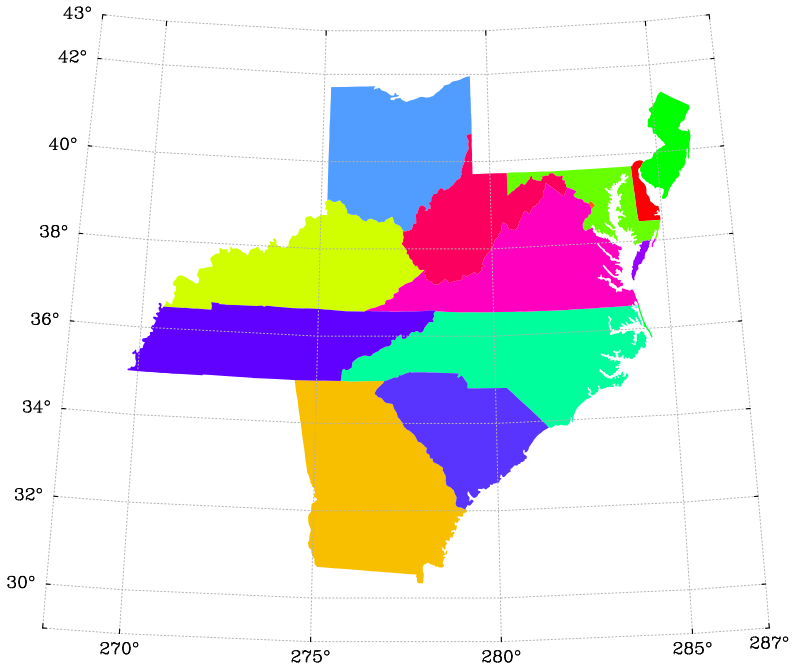

Figure 17: South Eastern US states with utm projection

```
> wAK = grep("Juneau",  us.cities$name[ wcaps]  )
> wcaps = wcaps[-wAK]
>         nam1 =  us.cities$name[wcaps]
>         lat1 =  us.cities$lat[wcaps]
>         lon1 =  us.cities$lon[wcaps]
>         pointsGEOmapXY(lat = lat1, lon = lon1, PROJ =proj, pch=8 )
> dev.off()
>
>
>
```

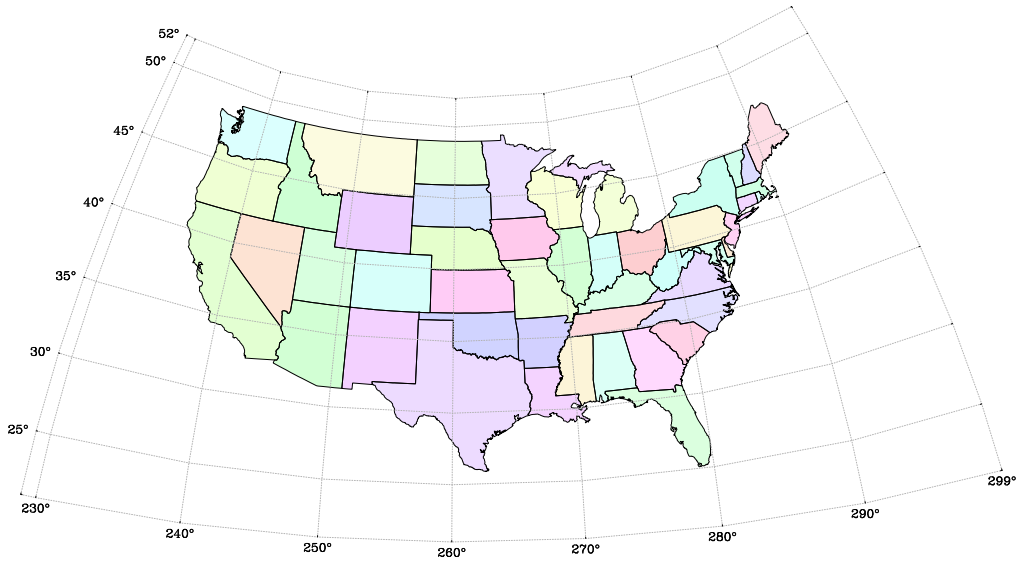

Figure 18: Pastel Color US map with transverse Mercator projection (Clark1866 datum)

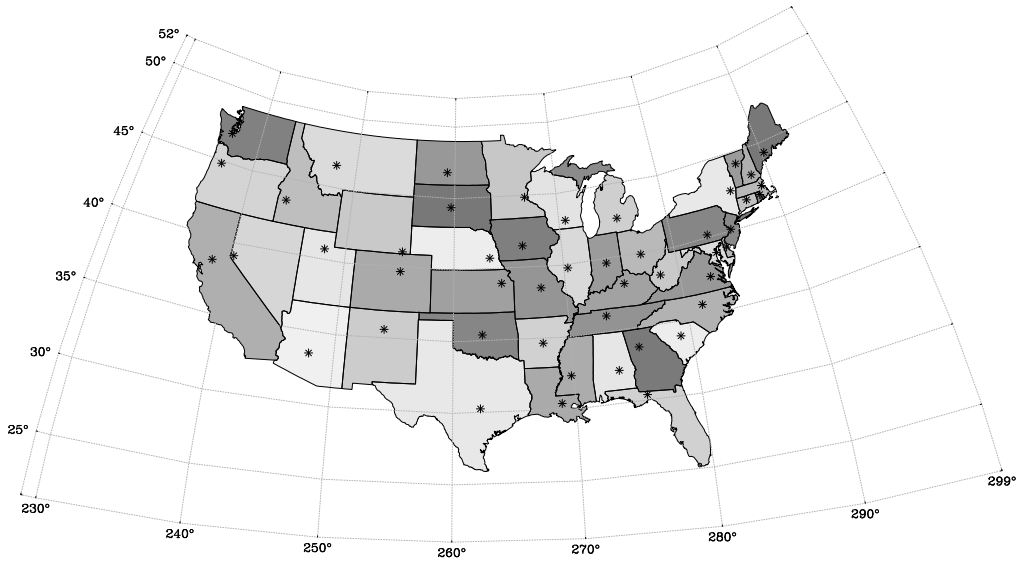

Figure 19: Black and White US with capitals