

# Cool Topo using GEOmap

Jonathan M. Lees  
University of North Carolina, Chapel Hill  
Department of Geological Sciences  
CB #3315, Mitchell Hall  
Chapel Hill, NC 27599-3315  
email: jonathan.lees@unc.edu  
ph: (919) 962-0695

This set of routines is intended to show how to make a striking topographic map using GEOmap, a package written in **R**. **R** and package GEOmap can be downloaded from the internet at:

<http://www.r-project.org/>.

```
> library(GEOmap)
> library(ETOPO)
> library(WORLDMAP2)
> data(world2)
> data(worldmap)
> data(ETOPO5)
> data(ETOPO2)

> PROJ = setPROJ(type = 1, LAT0 = 0, LON0 = 0)
> PLOC = list(lon = c(255, 294), lat = c(-36, 8))
> nn = names(PLOC)
> ilon = grep("lon", nn, ignore.case = TRUE)
> ilat = grep("lat", nn, ignore.case = TRUE)
> A = list(lat = PLOC[[ilat[1]]], lon = PLOC[[ilon[1]]], LAT = PLOC[[ilat[1]]],
+         LON = fmod(PLOC[[ilon[1]]], 360))
> ALOC = A
> TOPO = ETOPO2
> top2 = subsetTOPO(TOPO, ALOC)
> d1 = dim(top2$z)
> top2$z = top2$z[, d1[2]:1]
> image(top2, col = terrain.colors(100))
> calcol = settopocol()
> Cmat = TOPOCOL(top2, calcol$calcol)
> Dcol = attr(Cmat, "Dcol")
```

```

> X = GLOB.XY(top2$y[1], fmod(top2$x, 360), PROJ)
> DX = X$x[2] - X$x[1]
> Y = GLOB.XY(top2$y, top2$x[1], PROJ)
> any(diff(X$x) <= 0)
> any(diff(Y$y) <= 0)
> NEWimage = list(x = X$x, y = Y$y, z = top2$z/1000)

> postscript(file = "BIGmap_EXAMPLE.eps", width = 8, height = 10,
+   paper = "special", horizontal = FALSE, onefile = TRUE, print.it = FALSE)
> par(mai = c(0.2, 0.2, 0.2, 0.2))
> PMAT = persp(NEWimage$x, NEWimage$y, NEWimage$z, theta = 0, phi = 90,
+   r = 4000, col = Cmat[1:(Dcol[1] - 1), 1:(Dcol[2] - 1)], scale = FALSE,
+   ltheta = 120, lphi = 75, shade = 0.5, border = NA, expand = 200,
+   box = FALSE)
> sqrtICXY(NEWimage, PROJ, side = 1, LLgrid = FALSE, col = grey(0.7),
+   PMAT = PMAT)
> sqrtICXY(NEWimage, PROJ, side = 2, LLgrid = FALSE, col = grey(0.7),
+   PMAT = PMAT)
> dev.off()
> system("ps2png.prl BIGmap_EXAMPLE.eps")

```

## 1 Post Processing

If you are making a map in LINUX there is a very nice way to take the postscript output described here and convert it to a large png file suitable for powerpoint or other presentation programs.

```
epstopdf BIGmap_EXAMPLE.eps
```

```
gs -dBATCH -sDEVICE=png16m -dNOPAUSE -r300 -sOutputFile=BIGmap_EXAMPLE.png BIGmap_EXAMPLE.
```

This is wrapped together in a nice perl script

```

#!/usr/bin/perl
## convert a ps file to png

for($j=0; $j<=$#ARGV; $j++)
{

$file = $ARGV[$j];
$file =~ s/\.[eE]?[Pp][Ss]$//;

```

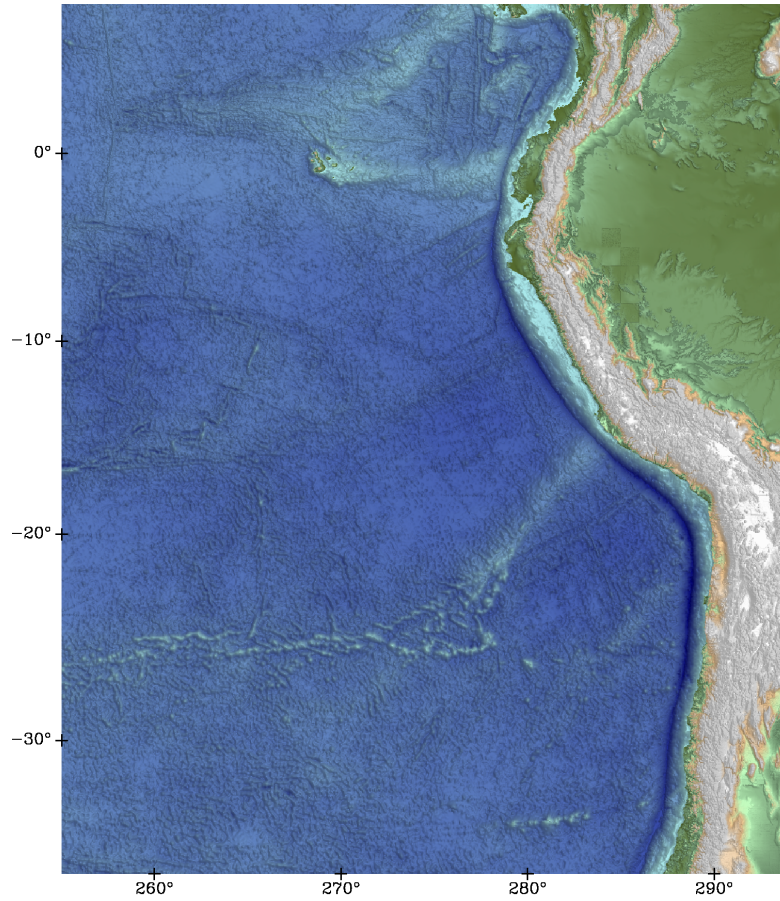


Figure 1: GEOMap Example **R**

```
$inpf = join(' ', $file, '.pdf');  
$outpf = join(' ', $file, '.png');  
  
system("epstopdf $ARGV[$j]");  
  
## here we scale it back to normal size  
system("gs -dBATCH -sDEVICE=png16m -dNOPAUSE -r300 -sOutputFile=$outpf $inpf");  
  
}
```

which can then be executed by typing:

```
ps2png.pl BIGmap_EXAMPLE.eps
```