HW10

In this homework you will make a GMT figure that shows yesterday’s earthquakes. This is based on the GMT v6 example\_22. It has been modified for GMT v5 (what is installed on the MACLAB computers). This kind of map has been a homework every year – and how to get the data automatically from the USGS changes every year!

We want to make the map from the GMT example\_22, but the map is missing something and we would like to fix it by adding the plate boundaries, colored by type. The scripts are annotated below in this file with my comments (red) and questions for you to answer (blue), the originals are in the folder.

example\_22 will be the base/main script – you will modify it (call it HW10.sh) to make the modified map. Paste in the necessary parts from the script plate\_bound\_map\_1.sh to make the final map. The final map should look like the map in HW10.sh.ps.

#!/bin/bash

# GMT EXAMPLE 22

#

# Purpose: Automatic map of yesterday’s world-wide seismicity

#mod Bob Smalley

#define output file name and clear old version (could make this better by checking for output file existence – will get rid of error message if it does not exit – do this).

ps=$0.ps

\rm $ps

#set some GMT environment variables

gmt set FONT\_ANNOT\_PRIMARY 10p FONT\_TITLE 18p FORMAT\_GEO\_MAP ddd:mm:ssF

#file name for yesterday’s quakes

file=usgs\_quakes\_22.txt

# Get the data (-s silently) from USGS using the curl)

#set minimum magnitude and the order you want the data returned

#you can find out how to do this and the options from the web page

# <https://earthquake.usgs.gov/fdsnws/event/1/>

MAG="minmagnitude=3"

ORDER="orderby=magnitude"

SITE="https://earthquake.usgs.gov/fdsnws/event/1/query.csv"

startdate=`date -v-1d +%F` #to get date a week ago date -v-1w +%F

enddate=`date +%F`

#need to use ASCII code “%20” for space here – don’t know why (observationally, spaces break it into 3 tokens and it messes things up. The original had the %20, I would not have figured it out – but now I and you know!)

TIME="starttime=${startdate}%2000:00:00&endtime=${enddate}%2000:00:00"

#form the command

URL="${SITE}?${TIME}&${MAG}&${ORDER}"

#now get the data using the command URL we formed curl

curl -s $URL > $file

# Count the number of events (to be used in title later. one less due to header)

n=`wc -l $file | nawk '{print $1-1}'`

#make new temporary file without header (could also put -h1 in gmt commands that use it.

tail -$n $file >$0.tmp

#rename it – this clobbers original file with headerless version.

mv $0.tmp usgs\_quakes\_22.txt

# Assign a string that contains the current user @ the current computer node.

# Note that two @@ is needed to print a single @ in gmt pstext:

#set me = "$user@@`hostname`"

me="GMT guru @@ GMTbox"

# Create standard seismicity color table (will color by depth)

#the color table is in the new file neis.cpt

gmt makecpt -Cred,green,blue -T0,100,300,10000 -N > neis.cpt

# Start plotting. First lay down base map, then plot quakes with size = magnitude \* 0.015" and color by depth:

touch $ps

gmt pscoast -Rg -JK180/9i -B45g30 -B+t"World-wide earthquake activity" –

#What does the Rg mean?

#What kind of projection is this?

Gburlywood -Slightblue \

-Dc -A1000 -K -Y2.75i >> $ps

#What does the “\” above do?

gmt psxy -R -J -O -K -Cneis.cpt -Sci -Wfaint -i2,1,3,4+s0.015 $file >> $ps

#What does the -I do?

#you may want to change the -Wfaint to -W0.1,black

# Create legend input file for NEIS quake plot

cat > neis.legend << END

H 16 1 $n events ge M3 during ${startdate}:00:00 to ${enddate}:00:00

D 0 1p

N 3

V 0 1p

S 0.1i c 0.1i red 0.25p 0.2i Shallow depth (0-100 km)

S 0.1i c 0.1i green 0.25p 0.2i Intermediate depth (100-300 km)

S 0.1i c 0.1i blue 0.25p 0.2i Very deep (> 300 km)

D 0 1p

V 0 1p

N 7

V 0 1p

S 0.1i c 0.06i - 0.25p 0.3i M 3

S 0.1i c 0.08i - 0.25p 0.3i M 4

S 0.1i c 0.10i - 0.25p 0.3i M 5

S 0.1i c 0.12i - 0.25p 0.3i M 6

S 0.1i c 0.14i - 0.25p 0.3i M 7

S 0.1i c 0.16i - 0.25p 0.3i M 8

S 0.1i c 0.18i - 0.25p 0.3i M 9

D 0 1p

V 0 1p

N 1

END

# Put together a reasonable legend text, and add logo and user's name:

cat << END >> neis.legend

G 0.25l

P

T USGS/NEIS most recent earthquakes for the last day.

T Data were obtained automatically from USGS Earthquake Hazards Program page

T @\_https://earthquake.usgs.gov@\_.

T Interested users may also receive email alerts from the USGS.

L 1 6 LB $me

END

# OK, now we can actually run gmt pslegend. We center the legend below the map.

# Trial and error shows that 1.7i is a good legend height:

gmt pslegend -DJBC+o0/0.4i+w7i/1.7i -R -J -O -F+p+glightyellow \

neis.legend >> $ps

#above is “too complicated to explain” – one reads the man page and figures out as much as possible and experiements.

# Clean up after ourselves:

#rm -f neis.\* gmt.conf usgs\_quakes\_22.txt

This is a file that plots plate boundaries colored by the type of boundary. It uses bash arrays to match the file names with colors programmatically (so you don’t have to hard code repeat stuff).

You will have to steal portions of the shell script below and move them to the other file (example\_22.sh) --- call the new file/program HW10.sh)

#!/bin/bash

PB=.

#What does the previous line do?

REGION=0/360/-60/60

PROJ=-Jm0.06

RED=255/0/0

BLUE=0/0/255

GREEN=0/255/0

MAGENTA=255/0/255

CYAN=0/255/255

MOREPS=-K

CONTINUEPS="-K -O"

ENDPS=-O

OUTFILE=$0.ps

X=-X2.8

Y=-Y3.6

#for consecutive index numbering can use

#set up two arrays – one with the boundary identification component of the file name and one with the corresponding colors for drawing the boundary.

BTYPE=( "Spreading" "Transform" "Convergent" "Subduction" )

BCOLOR=( "red" "green" "blue" "purple" )

#now build the plate boundary data file names – look at the file names

#to see what we are trying to generate

#notice the indexing [${cnt}] and recovering ${BTYPE[${cnt}]} of array data

#using arrays since need 2 (or more) matched sets of data – file types and

#color in this example

fn=0

for ((cnt=0; cnt<=3; cnt++))

do

files=`ls $PB/\*${BTYPE[${cnt}]}\*gmt`

for file in $files

do

#save the full file names and colors in new arrays

fname[++fn]=$file

fcolor[fn]=${cnt}

done

done

#now start plotting

pscoast -R$REGION $PROJ -G200 -B30g15 -W1 $MOREPS $X $Y -Dh -U > $OUTFILE

for ((cnt=1; cnt<=$fn; cnt++))

do

color=${BCOLOR[${fcolor[${cnt}]}]}

echo file num $cnt, name ${fname[${cnt}]}, has color $color

psxy -R$REGION $PROJ -M$ -W1/${BCOLOR[${fcolor[${cnt}]}]} $CONTINUEPS -V ${fname[${cnt}]} >> $OUTFILE

#explain/give value for the W flag components and the input file name above

done

echo showpage >> $OUTFILE

#what does the above line do?