By the end of this tutorial you will be able to create the following figures:
By the end of this tutorial you will be able to create the following figures:

**Large Earthquakes of Southeast Asia**

**Earthquakes Greater than Magnitude 5**

- **Magnitude**
  - 5
  - 6
  - 7
  - 8
  - 9

- **Depth (km)**
  - 0–25
  - 25–50
  - 50+
Map Projections

Plot the features of a *spherical* Earth on a *flat* surface

Mercator

Cylindrical Equal Area

Cassini

Plot the features of a *spherical* Earth on a *flat* surface

!!!! All projections distort angles, areas, or both !!!!

Winkel

Robinson

Mollweide

Review of Some Unix Terms

**Naming files:** Use only alphanumeric characters, “-”, and “_”. Don’t use spaces or any other characters (*&%$!^\~) when naming a file since they all have a special meaning in Unix. Unix is case sensitive so UPPER CASE filenames are different from lower case ones. Also, try to keep filenames short so there is less to type.

**Command:** The name of a program. When you press enter, the computer performs an action.

**Options:** These are specified after the command, and are usually indicated by a dash. They modify the behavior of the command to make it do what you want. See the command “man” page (manual pages) to learn more about the capabilities of each command.

**Command line:** The prompt on an active terminal where you can write a line of code. When you want to execute multiple commands, put them into a script (text file) to save yourself time typing.

**Path:** List of directories that your shell (the program that interprets the commands) searches when you enter a command.

**Directory:** Same as folders. Everything you use in one script must be in the same directory, or in your path.
Unix Special Characters

Redirection of program inputs and outputs:
>   e.g. file1 > file2 file1 overwrites file2
>> e.g. file1 >> file2 file1 appends file2
<   e.g. file1 < file2 file1 takes input from file2
|   pipe. sends output to next command as input

Directory specifications:
.   current directory
..  parent directory
~   shorthand for the path to the home directory

Comments and continuations:
#   denotes a comment; any characters following will be ignored
\  ignores a carriage return, useful for multi-line commands in a script
Introduction to `psbasemap`

```
gmt psbasemap -JM10c -R80/110/0/35 -Ba10 > map1a.ps
```

- `gmt psbasemap` All GMT commands using GMT5 must be started with `gmt`. The tool `psbasemap` produces a base map, as the name suggests.

- `-J -R -B` A single dash starts an option (sometimes called a flag). Some options are required, others are added for controlling the output. All options must be separated by spaces. Some options also allow specifications, e.g. `–JM10c`

- `> map1a.ps` Redirect the output of the `psbasemap` command into a text file called `map1a.ps`
Introduction to psbasemap

gmt psbasemap -JM10c -R80/110/0/35 -Ba10 > map1a.ps

-JM10c The -J option specifies the type of map projection. The M indicates Mercator. When upper case, you specify the map width afterwards. The 10c indicates 10 cm. If you would like you can use i for inches.

-R80/110/0/35 The -R option specifies the map limits, with the format: west/east/south/north. Longitude for west & east and latitude for north & south.

-Ba10 The -B option specifies map boundary annotations. The a indicates you are specifying how often (every 10 units, degrees in this case) there should be an annotation.
Introduction to psbasemap

```
gmt psbasemap -JM10c -R80/110/0/35 -Ba10 > map1a.ps
```

On a Mac, we can open the PostScript file directly from the terminal instead of going to the folder in the Finder. Simply type

```
open map1a.ps
```

to open it with the default program.

*NOTE: This applies only to linux or mac users. To view the postscript file on a pc, drag the icon onto ghostview.*
Introduction to psbasemap

```bash
gmt psbasemap -JM10c -R80/110/0/35 -Ba10 -P > map1a.ps
```

-P If your map is appearing sideways add in the -P option. This option selects the portrait plotting mode. The default is landscape orientation.
# change -Ba10 to -Bg10, map1a.ps to map1b.ps

gmt psbasemap –JM10c –R80/110/0/35 –Bg10 > map1b.ps

# The first line starts with a pound sign. Everything that comes after a pound sign is a comment and will not be interpreted as a command. Use comments to keep track of your work to remember what you’ve done, and so someone else can read your code.

-Bg10 We switched the a to g. The g indicates gridline plotting (every 10 degrees). Since we removed the a we no longer have any boundary annotations.

If you haven’t yet, you should start putting all your commands into scripts. For details, see our Unix tutorial.
Introduction to psbasemap

gmt psbasemap -JM10c -R80/110/0/35 -Ba10g10 > map1c.ps

-Ba10g10 Here we combine the a and g options to plot annotations and gridlines. Since both have a 10 afterwards, both are plotted every 10 degrees.
Introduction to `psbasemap`

```
gmt psbasemap -JM10c -R80/110/0/35 -Bxa10g10 -Bya5g5 > map1d.ps
```

`-Bxa10g10 -Bya5g5` What if we don’t want the annotations and gridlines to be the same for both latitude and longitude? To define them independently, we divide the `-B` option into x and y components using `-Bx` and `-By`. The annotation and gridline syntax is the same as before, but now they can have different values. This can be especially useful for maps with large aspect ratios, near-polar maps, or unusual projections.

This should all be on one line.
Introduction to psbasemap

gmt psbasemap -JM10c -R80/110/0/35 -Bxa10g10 -Bya5g5 -BWesN > maple.ps

-Bxa10g10 -Bya5g5 -BWesN Finally, we can define what sides to plot the annotations with -BWesN. This specifies which edges (West, east, south, North) we would like to have longitude or latitude annotations. The capital letters W & N indicate which edges we want annotated, and the lowercase letters e & s indicate that we want to draw the east and south edges they will be left blank.
Introduction to \textit{pscoast}

\begin{verbatim}
gmt pscoast -JM10c -R80/110/0/35 -W0.5p -K > map2a.ps

gmt psbasemap -JM10c -R80/110/0/35 -Bxa10g10 -Bya5g5 -BWesN -O >> map2a.ps
\end{verbatim}

\texttt{gmt pscoast} The \texttt{pscoast} command draws coastlines (surprise!) and a few other geographical features such as country or state boundaries.

\textit{Note: coastline datasets are included in the GMT package, so you do not need to find these separately unless you messed something up during installation.}

\texttt{--K} We added this option to \texttt{pscoast}, indicating that we are adding more layers on after this command. Every command should include the \texttt{--K} option except the very last one.
Introduction to pscoast

gmt pscoast -JM10c -R80/110/0/35 -W0.5p -K > map2a.ps
gmt psbasemap -JM10c -R80/110/0/35 -Bxa10g10 -Bya5g5 -BWesN -O >> map2a.ps

-O Notice that we added an option to the psbasemap command. The “O” (letter O, not number 0) stands for “overlay”, indicating that the features in this layer will be overlain on top of the previous layer instead of starting a new plot. -O should be included in every GMT line except for the first.

>> Instead of using the single redirect symbol > we used the double redirect symbol >>. A single > overwrites the file with the output from the command, while the double >> appends the program output to the file.
Introduction to pscoast

```bash
gmt pscoast -JM10c -R80/110/0/35 -W0.5p -K > map2a.ps
gmt psbasemap -JM10c -R80/110/0/35 -Bxa10g10 -Bya5g5 -BWesN -O >> map2a.ps
```

- **JM10c** - **R80/110/0/35** Both `pscoast` and `psbasemap` use the same projection and limits options. We want to use the same options so the base map and the coastlines will line up correctly.

- **W0.5p** The `-W` option controls the drawing of a coastline. The number afterwards specifies the thickness of those lines. Here, **0.5p** means draw a line half a point thick, where a point is 1/72 of an inch (the `p` indicates a measurement in points).
Introduction to pscoast

gmt pscoast -JM10c -R80/110/0/35 -Wlp -L87/3+c3+w1000k+u -K > map2b.ps

-L87/3+c3+w1000k+u The -Llon/lat option creates a map scale centered at lon/lat (in this example, 87/3). Most GMT options have “sub-options” which allow for fine tuning the behavior. These are usually called with a + symbol. Here, the +c3 sub-option indicates that the scale is calculated at a latitude of 3°N. The sub-option +w1000k defines the length of the scale bar, with k indicating km. Finally, +u indicates to include the units on the scale bar.

Option to add a scale bar
Introduction to pscoast

```
gmt pscoast -JM10c -R80/110/0/35 -W1p -L87/3+c3+w500k+u -G255/255/0
-K > map2b.ps
```

-G255/255/0 The –G option colors in dry areas, i.e. land. The color is defined by RGB (red/green/blue) values; each value can range from 0 to 255 (2^8 possibilities for each). Yellow is 255/255/0. GMT also recognizes many color names; in this case, we could have written –Gyellow and gotten the same result.

To explore RGB colors, check out this link:
http://www.rapidtables.com/web/color/RGB_Color.htm
Or search for “rgb colors”
Introduction to pscoast

```
gmt pscoast -JM10c -R80/110/0/35 -W1p -L87/3+c3+w500k+u -G255/255/0 -S50/100/255 -K > map2b.ps
```

- Option to color wet areas (water)

-S50/100/255 Fills in the wet areas, i.e. oceans, lakes, and rivers. The syntax is the same as in the -G option.
Introduction to pscoast

gmt pscoast -JM10c -R80/110/0/35 -W1p -L87/3+c3+w500k+u -G255/255/0 -S50/100/255 -N1/0.25p -Dc -K > map2b.ps

Draw country boundaries
Define coastline resolution

-N1/0.25p The -N option draws political boundaries. The number directly afterwards specifies the type of boundary; -N1 draws country boundaries. The number after the slash indicates the line thickness of the boundaries.

-Dc Choose the resolution of the coastline dataset, where the options are (f)ull, (h)igh, (i)ntermediate, (l)ow, and (c)rude. The default resolution is i. Lower resolutions make smaller file sizes, and are more appropriate for maps showing large regions, whereas high resolutions are better for showing details of zoomed-in areas.
Introduction to pscoast

```
gmt pscoast -JM10c -R80/110/0/35 -Wlp -L87/3+c3+w500k+u -G255/255/0
-S50/100/255 -N1/0.25p -Di -K > map2b.ps
```

- **Di** Intermediate resolution looks pretty good for our map region.
Map of Asia

Now we are going to make a slightly different map of Asia using `pscoast` and `psbasemap`. We will use the relevant GMT options as well as shell scripting tools.

When you are finished, you will have the map shown at the right.
#!/bin/sh

PROJ="-JC15c"
LIMS="-R50/170/-10/70"
BLUE="50/100/255"
SHIFT="-X8c -Y4c"
PSFILE="asia.ps"

gmt pscoast $PROJ $LIMS -W1p -Gyellow -S$BLUE -N1,0.25p $SHIFT -Dc -K > $PSFILE

gmt psbasemap $PROJ $LIMS -Bxa10g10 -Bya5g5 -BWeSn -O >> $PSFILE
Map of Asia

Recall the use of variables from our Unix tutorial. They help organize, they save space, and make it so you only have to change a value in one place rather than in every instance in your script.

Define some important variables.

```bash
#!/bin/sh
PROJ="-JC15c"
LIMS="-R50/170/-10/70"
WET="175/200/225"
DRY="225/200/175"
PSFILE="asia.ps"
gmt pscoast $PROJ $LIMS -W1p -Dc -G$DRY -S$WET -N1/0.5p -K > $PSFILE
gmt psbasemap $PROJ $LIMS -Bxa10g10 -Bya5g5 -BWeSn -O >> $PSFILE
```
Map of Asia

#!/bin/sh
PROJ="-JC15c"
LIMS="-R50/170/-10/70"
WET="175/200/225"
DRY="225/200/175"
PSFILE="asia.ps"
gmt psoast $PROJ $LIMS -W1p -Dc -G$DRY -S$WET -N1/0.5p -K > $PSFILE
gmt psbasemap $PROJ $LIMS -Bxa10g10 -Bya5g5 -BWeSn -O >> $PSFILE

Map projection: the -JC indicates the Cassini cylindrical projection
Map of Asia

#!/bin/sh

PROJ="-JC15c"
LIMS="-R50/170/-10/70"
WET="175/200/225"
DRY="225/200/175"
PSFILE="asia.ps"

gmt pscost $PROJ $LIMS -W1p -Dc -G$DRY -S$WET -N1/0.5p -K > $PSFILE

gmt pbsbasemap $PROJ $LIMS -Bxa10g10 -Bya5g5 -BWeSn -O >> $PSFILE

Map limits: in this case, we have chosen a much larger range
Map of Asia

#!/bin/sh

PROJ="-JC15c"
LIMS="-R50/170/-10/70"
WET="175/200/225"
DRY="225/200/175"
PSFILE="asia.ps"

gmt pscoast $PROJ $LIMS -W1p -Dc -G$DRY -S$WET -N1/0.5p -K > $PSFILE

gmt psbasemap $PROJ $LIMS -Bxa10g10 -Bya5g5 -BWeSn -O >> $PSFILE

The color for wet areas
Map of Asia

#!/bin/sh
PROJ="-JC15c"
LIMS="-R50/170/-10/70"
WET="175/200/225"
DRY="225/200/175"
PSFILE="asia.ps"
gmt pscoast $PROJ $LIMS -W1p -Dc -G$DRY -S$WET -N1/0.5p -K > $PSFILE
gmt psbasemap $PROJ $LIMS -Bxa10g10 -Bya5g5 -BWeSn -O >> $PSFILE

The color for dry areas
Map of Asia

#!/bin/sh

PROJ="-JC15c"
LIMS="-R50/170/-10/70"
WET="175/200/225"
DRY="225/200/175"
PSFILE="asia.ps"

gmt pscoast $PROJ $LIMS -W1p -Dc -G$DRY -S$WET -N1/0.5p -K > $PSFILE

gmt psbasemap $PROJ $LIMS -Bxa10g10 -Bya5g5 -BWeSn -O >> $PSFILE

The name of the output PostScript file.
Map of Asia

#!/bin/sh
PROJ="-JC15c"
LIMS="-R50/170/-10/70"
WET="175/200/225"
DRY="225/200/175"
PSFILE="asia.ps"
gmt pscoast $PROJ $LIMS -W1p -Dc -G$DRY -S$WET -N1/0.5p -K > $PSFILE
gmt psbasemap $PROJ $LIMS -Bxa10g10 -Bya5g5 -BWeSn -O >> $PSFILE

Run your script....
Plotting x-y data
Start with a familiar script.  
This generates the coastline and base map.
Introduction to psxy

```
gmt psxy stars.xy $PROJ $LIMS -Sa0.3i -W0.25p -Gred -O -K >> $PSFILE
```

**psxy** This command plots x-y data as lines or symbols.

**stars.xy** The `psxy` command requires x-y input to know where to plot the points. Our data must have the first column be longitude and the second be latitude (*Note: at some point you will flip these; always check your input file*).

Create a text file named **stars.xy** containing an x-y coordinate in the map area. Save it in the same folder as your script. Try the coordinates of Bangkok:

100.494 13.752
Introduction to psxy

```bash
gmt psxy stars.xy $PROJ $LIMS -Sa0.3i -W0.25p -Gred -O -K >> $PSFILE
```

`-Sa0.3i` The default behavior of `psxy` is to draw lines connecting the coordinates in the file. To draw symbols, we use the `-s` option. There are many different types of symbols you can use; to see these look at the `psxy` man page. `--Sa0.3i` specifies drawing a star that fits into a circle 0.3 inches in diameter.
Introduction to psxy

```bash
gmt psxy stars.xy $PROJ $LIMS -Sa0.3i -W0.25p -Gred -O -K >> $PSFILE
```

-W0.25p -Gred To outline and color the symbols, we use the -W and -G options. -W defines the pen used to outline the symbol, and -G specifies the color to fill the symbol. At least one of these options must be used. If no fill option is selected, the symbol is plotted transparent.
Introduction to psxy

gmt psxy stars.xy $PROJ $LIMS -Sa0.3i -W0.25p -Gred -O -K >> $PSFILE

$PROJ $LIMS We use the same projection and limits variables as the other commands.

-K -O Since we are adding another layer with psbasemap after this one, we must use the -K option. Likewise, since this layer is overlain on the previous pscoast layer, we must have -O.
Plotting earthquake data using psxy

Now we will add some earthquake locations to this figure. You can download a text file with earthquakes at:

https://geodyn.psu.edu/tutorials/eqs.xy

This text file contains magnitude 5 and greater earthquakes in the vicinity of the mapped region from 1996 to 2006.
Plotting earthquake data using psxy

```
gmt psxy eqs.xy $PROJ $LIMS -Sc0.15i -W0.5p -Ggreen -O -K $PSFILE
```

- **-Sc0.15i** To plot the earthquakes as circles, use `-Sc`. The diameter of the symbols is 0.15 inches.

You can set the fill to any color you choose (since we are going to change it soon), but here it is set to green with `-Ggreen`. 
Plotting earthquake data using psxy

```
gmt psxy eqs.xy $PROJ $LIMS -Sc0.15i -W0.5p -Ggreen -O -K >> $PSFILE
```

- **Sc0.15i** To plot the earthquakes as circles, use `-Sc`. The diameter of the symbols is 0.15 inches.

You can set the fill to any color you choose (since we are going to change it soon), but here it is set to green with `-Ggreen`.

*This map is not particularly useful; it only contains information about earthquake location but nothing about depth or magnitude. Next, we will add that information to the map.*
Plotting earthquake data using psxy

```bash
gmt psxy eqs.xy $PROJ $LIMS -Sc0.15i -W0.5p -Ggreen -O -K >> $PSFILE
```

- **-Sc0.15i** To plot the earthquakes as circles, use `-Sc`. The diameter of the symbols is 0.15 inches.

You can set the fill to any color you choose (since we are going to change it soon), but here it is set to green with `-Ggreen`.

This data is already conveniently in the data file. The columns are:

- longitude
- latitude
- depth
- magnitude
Plotting earthquake data using psxy

```bash
gmt psxy eqs.xy $PROJ $LIMS -Sc -W0.5p -Ceq.cpt -O -K >> $PSFILE
```

`-Ceq.cpt` The `-C` option tells `psxy` you want to color your symbols based on the value of the 3rd column of data, using the scheme defined in the color palette file `eq.cpt`. In this case, the 3rd column of our file contains earthquake depth, which is how we want to color our symbols. A good earthquake depth color palette file might look like this (save it as `eq.cpt`):

```
0  255/0/0   10  255/0/0
10 255/155/0 25  255/155/0
25 0/255/0   50  0/255/0
50 0/0/255   100 0/0/255
B 255/0/0
F 0/0/255
```
Plotting earthquake data using psxy

```
gmt psxy eqs.xy $PROJ $LIMS -Scc -W0.5p -Ceq.cpt -O -K >> $PSFILE
```

-Ceq.cpt The -C option tells psxy you want to color your symbols based on the value of the 3rd column of data, using the scheme defined in the color palette file eq.cpt. In this case, the 3rd column of our file contains earthquake depth, which is how we want to color our symbols. A good earthquake depth color palette file might look like this (save it as eq.cpt):

```
0  255/0/0   10  255/0/0
10 255/155/0 25  255/155/0
25 0/255/0   50  0/255/0
50 0/0/255   100 0/0/255
B 255/0/0
F 0/0/255
```

Color range: 0-10
Plotting earthquake data using psxy

```
gmt psxy eqs.xy $PROJ $LIMS -Sc1 -W0.5p -Ceq.cpt -O -K >> $PSFILE
```

**-Ceq.cpt** The **-C** option tells **psxy** you want to color your symbols based on the value of the 3\textsuperscript{rd} column of data, using the scheme defined in the color palette file **eq.cpt**. In this case, the 3\textsuperscript{rd} column of our file contains earthquake depth, which is how we want to color our symbols. A good earthquake depth color palette file might look like this (save it as **eq.cpt**):

```
0 255/0/0 10 255/0/0
10 255/155/0 25 255/155/0
25 0/255/0 50 0/255/0
50 0/0/255 100 0/0/255
B 255/0/0
F 0/0/255
```

Starting color: 255/0/0
Ending color: 255/0/0
Plotting earthquake data using psxy

gmt psxy eqs.xy $PROJ $LIMS -Scc -W0.5p -Ceq.cpt -O -K $PSFILE

-Ceq.cpt The -C option tells psxy you want to color your symbols based on the value of the 3rd column of data, using the scheme defined in the color palette file eq.cpt. In this case, the 3rd column of our file contains earthquake depth, which is how we want to color our symbols. A good earthquake depth color palette file might look like this (save it as eq.cpt):

<table>
<thead>
<tr>
<th>Depth</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>255/0/0</td>
</tr>
<tr>
<td>10</td>
<td>255/0/0</td>
</tr>
<tr>
<td>25</td>
<td>255/155/0</td>
</tr>
<tr>
<td>50</td>
<td>0/255/0</td>
</tr>
<tr>
<td>100</td>
<td>0/0/255</td>
</tr>
<tr>
<td>B</td>
<td>255/0/0</td>
</tr>
<tr>
<td>F</td>
<td>0/0/255</td>
</tr>
</tbody>
</table>

Color slices must be continuous.
Plotting earthquake data using psxy

gmt psxy eqs.xy $PROJ $LIMS -Scc -W0.5p -Ceq.cpt -O -K >> $PSFILE

-Ceq.cpt The -C option tells psxy you want to color your symbols based on the value of the 3rd column of data, using the scheme defined in the color palette file eq.cpt. In this case, the 3rd column of our file contains earthquake depth, which is how we want to color our symbols. A good earthquake depth color palette file might look like this (save it as eq.cpt):

<table>
<thead>
<tr>
<th>Depth</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>255/0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>255/155/0</td>
<td>25</td>
<td>255/155/0</td>
</tr>
<tr>
<td>25</td>
<td>0/255/0</td>
<td>50</td>
<td>0/255/0</td>
</tr>
<tr>
<td>50</td>
<td>0/0/255</td>
<td>100</td>
<td>0/0/255</td>
</tr>
<tr>
<td>B</td>
<td>255/0/0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>0/0/255</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Background and foreground colors
Plotting earthquake data using psxy

```bash
gmt psxy eqs.xy $PROJ $LIMS -ScC -W0.5p -Ceq.cpt -O -K >> $PSFILE
```

*Change symbol size*

`-ScC` Instead of explicitly defining the size of the symbol in the `-S` option, we do not specify a value here, only a unit `c` (for cm). Doing this scales the symbol size to be the value of the size column in cm.

If a color palette is defined (as it is here with `-Ceq.cpt`), then the color value *always* goes into the 3\textsuperscript{rd} column, immediately after the x and y values of the point. Then the size is the value in the 4\textsuperscript{th} column.

If you do not define a color palette, then the value in the 3\textsuperscript{rd} column of the input is taken as the size.

To manipulate columns of files, use awk (see our tutorial online!).
At this point, your script should look something like this:

```sh
#!/bin/sh
PROJ="-JM15c"
LIMS="-R80/110/0/35"
PSFILE="earthquakes.ps"
gmt pscoast $PROJ $LIMS -W1p -Dc -N1/0.5p -K > $PSFILE
gmt psxy stars.xy $PROJ $LIMS -Sa0.3i -W0.25p -Gred -O -K >> $PSFILE
gmt psxy eqs.xy $PROJ $LIMS -Sc -W0.5p -Ceq.cpt -O -K >> $PSFILE
gmt psbasemap $PROJ $LIMS -Bxa10g10 -Bya5g5 -BWeSn -O >> $PSFILE
```