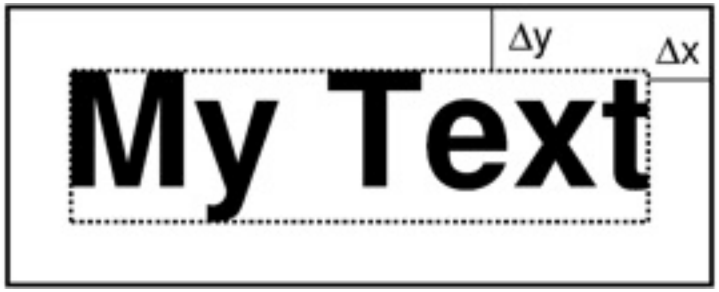
Plotting text lines and paragraphs

- In addition to **–R**, **–J**, **–B** etc., options are:

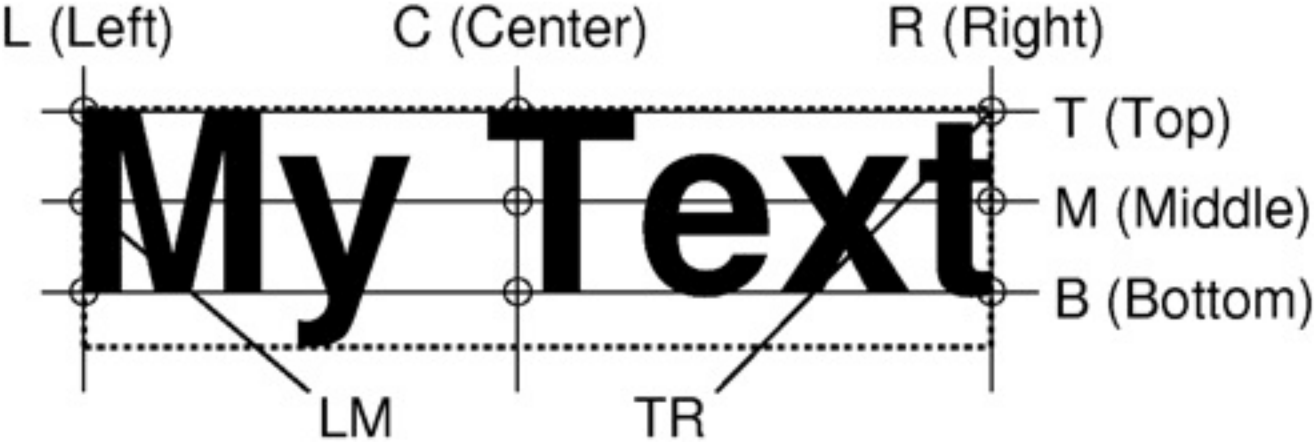
Option	**Purpose**
–Cdx/dy	Spacing between text and the text box (see **–W**)
–Ddx/dy	Offsets text from specified point location
–Gfill	Sets the color of the text
–L	Lists the font ids and exits
–M	Paragraph mode [Default is text line mode]
–N	Deactivates clipping at the borders
–Spen	Selects outline font and sets pen attributes
–W[fill][**o**|**O**|**c**|**C**[pen]]	Paint text box; draw outline by appending **o** (also see **–C**)



pstext text line input and setup

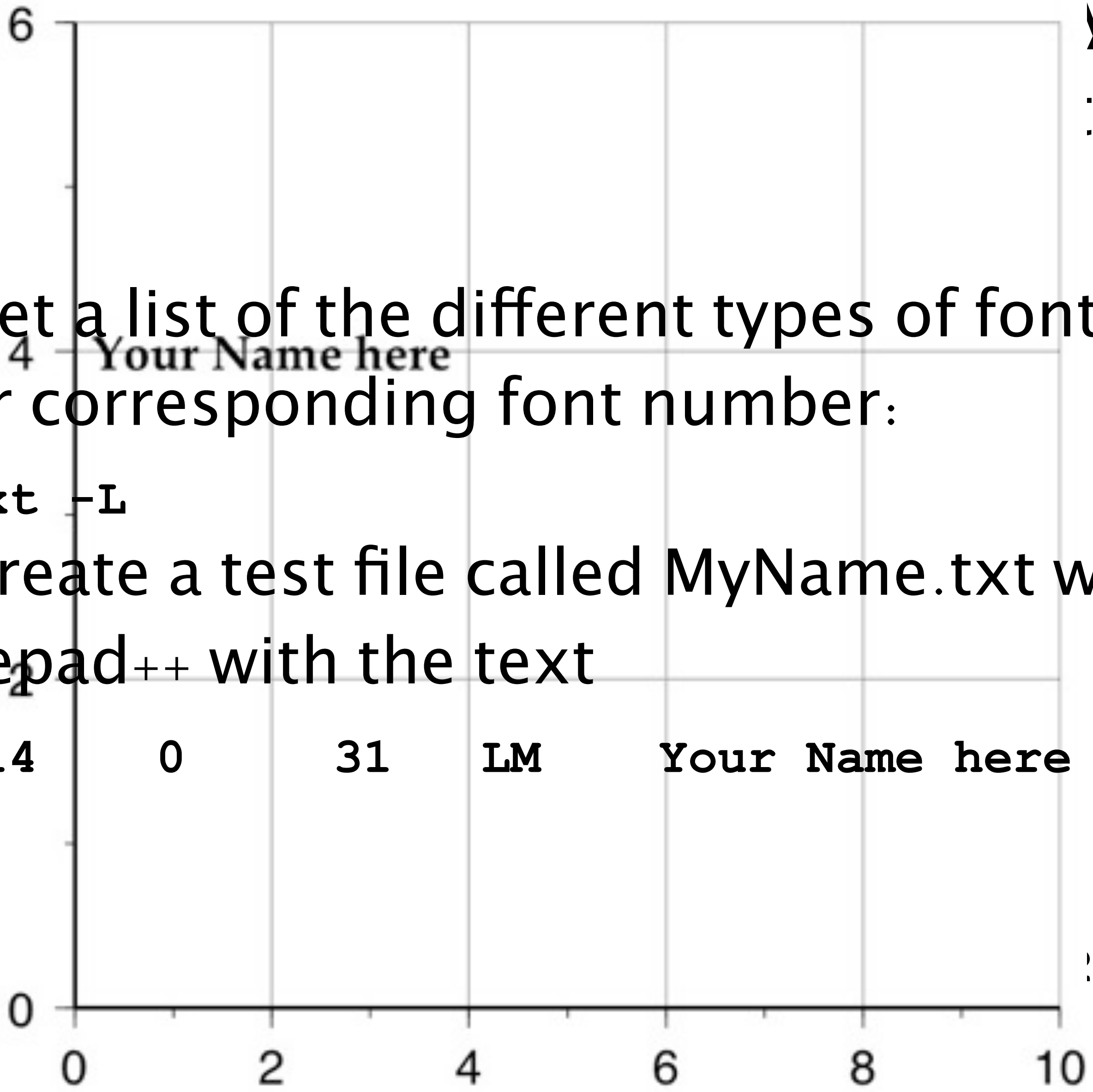
x y size angle font justify text

- x y is map coordinates
- size is font size in points
- angle is inclination of text baseline w.r.t. horizontal
- font is font name or number (see **–L**)
- justify relates (x, y) to a point on the text string
- text



Exercise: Plot your name

- Crea your nam type of Pala



- 1) Get a list of the different types of fonts and their corresponding font number:

```
pstext -L
```

- 2) Create a test file called MyName.txt with Notepad++ with the text

```
2 4 14      0      31     LM      Your Name here
```

x y s

```
pstext                                                  WS -P >
ex12.p
```

Exercise: Plot your name cont…

- You can also make the same plot using the command line

```
pstext -R0/10/0/6 -JX10 -Ba2f1g2WS -P <<end > ex12.ps
2 4 14 0 31 CM Your Name here
end
```

- Plot your name using font size of 20 and a dark green Helvetica-Oblique font

- Plot your name in magenta using whichever font and font size you like but the text must be vertical

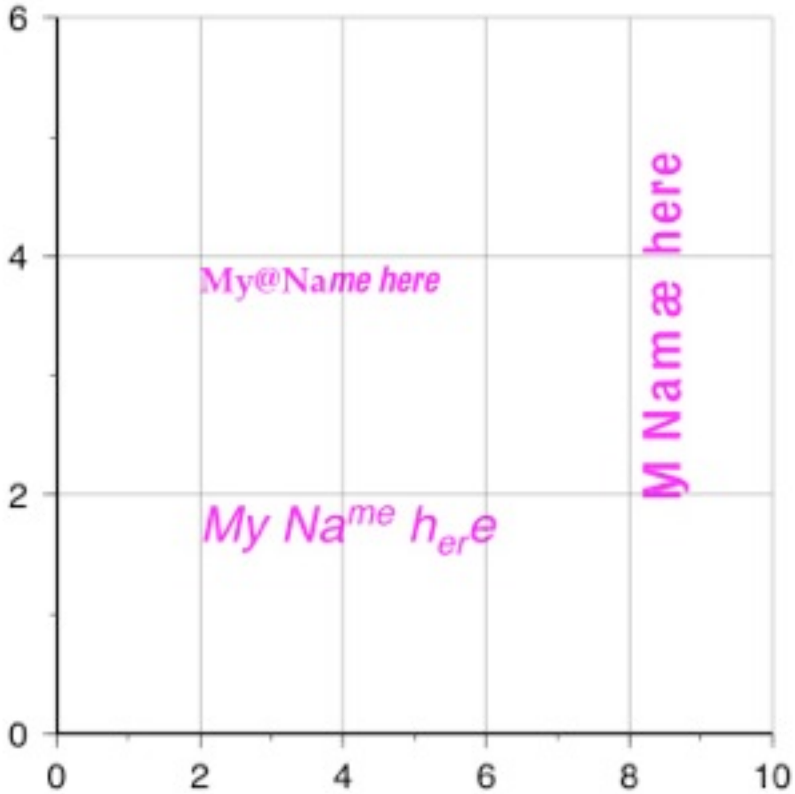
GMT text escape sequences

Code	Effect
@~	Turns symbol font on or off
@%font%	Change font; @%% resets to past font
@+	Turns superscript on or off
@-	Turns subscript on or off
@#	Turns small caps on or off
@!	Creates composite of next two characters
@@	Prints the @ sign itself
@E or @e	Æ or æ
@O or @o	Ø or ø
@A or @a	Å or å

For Example

MyName.txt looks like:

```
2 4 14 0 31 LT My@@Na@%24%me here
2 2 20 0 2 LT My Na@+me@+ h@-er@-e
8 2 20 90 15 LT @!My Nam@e here
```



Plotted with

```
pstext -R0/10/0/6 -JX10 -Ba2f1g2WS -P -Gmagenta MyName.txt
> ex12.ps
```

Or

```
pstext -R0/10/0/6 -JX10 -Ba2f1g2WS -P -Gmagenta MyName.txt
<<end > ex12.ps
2 4 14 0 31 LT My@@Na@%24%me here
2 2 20 0 2 LT My Na@+me@+ h@-er@-e
8 2 20 90 15 LT @!My Nam@e here
end
```

pstext paragraph mode (-M)

- Used to typeset justified text such as figure captions
- Sub-headings must indicate paragraph parameters:
  - < x y size angle font justify linespace parwidth parjust
  - parjust can be l, c, r, j
  - Sub-header must be followed by one or more text lines
- Takes 3 additional escape sequences:

Code	Effect
@;r/g/b;	Change font color, @;; resets it
@:size:	Change font size, @:: resets it
@_	Turn underline on or off

Example of pstext in Paragraph mode

Command looks like:

```
pstext -R0/10/0/6 -JX10 -Ba2f1g2WS -P -Gmagenta
MyParagraph.txt -M -V > MyParagraph.ps
```

MyParagraph.txt looks like

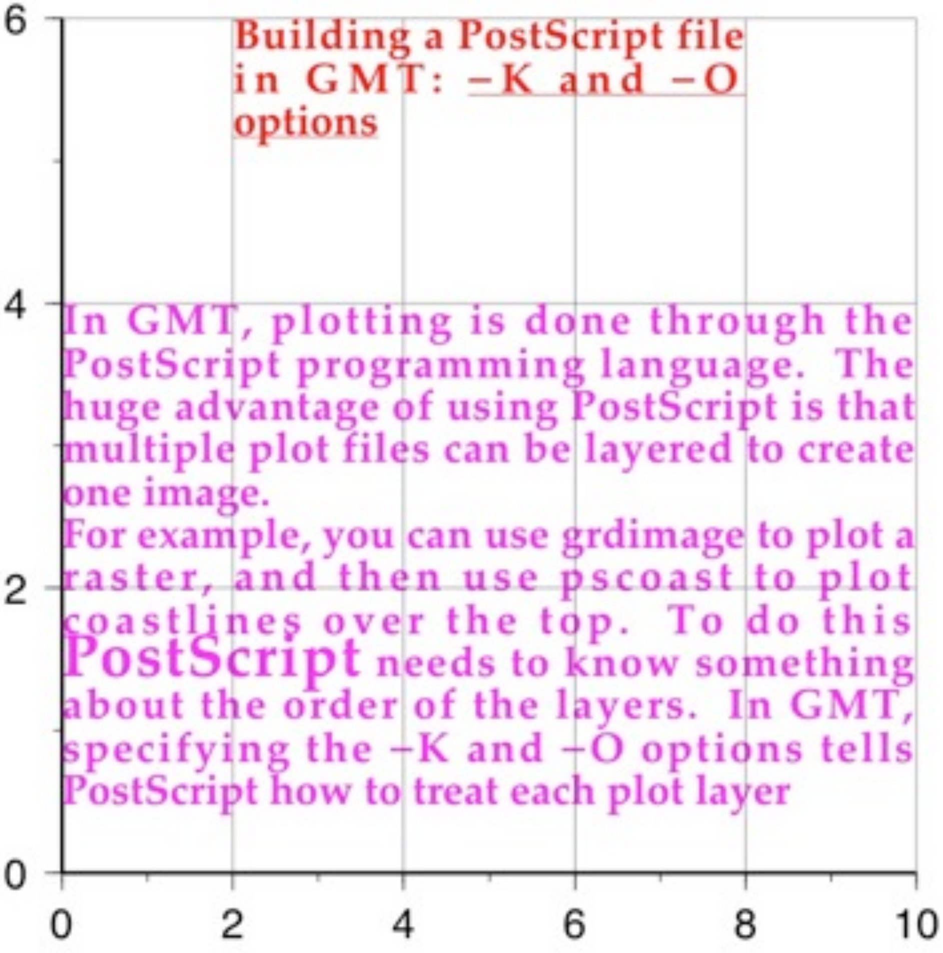
```
> 2 6 14 0 31 LT 0.5 6 j

@;255/0/0;Building a PostScript
options@_@;;

> 0 4 14 0 31 LT 0.5 10 j

In GMT, plotting is done throug
language. The huge advantage of
multiple plot files can be laye


For example, you can use grdima
use pscoast to plot coastlines
20:PostScript@:: needs to know
the layers. In GMT, specifying
PostScript how to treat each p
```



Building a PostScript file
in GMT: –K and –O
options

In GMT, plotting is done through the PostScript programming language. The huge advantage of using PostScript is that multiple plot files can be layered to create one image.
For example, you can use grdimage to plot a raster, and then use pscoast to plot coastlines over the top. To do this PostScript needs to know something about the order of the layers. In GMT, specifying the –K and –O options tells PostScript how to treat each plot layer

Exercise: Global Cities

- Task: Make a global Robinson Map
- Create a file with the long, lat locations of 4 cities. (Get locations from wcity_major.dat in your tutorial directory downloaded from ftp.geosci.usyd.edu.au/pub/marias/tutorial)
- Plot red squares for 4 cities. Label the cities.
- Place justified figure caption in a white, outlined box in the lower left corner
- Caption can be whatever but should take up more than 2 lines, start with "**Figure 1.**" and exercise the underline mechanism a bit
- Don't be afraid to look up man pstext

```
#!/bin/ksh
# Project: Global Cities Exercise
# Date:
# Author:              Jo Whittaker

pscoast -Rd (or -Rg) -JN.......        -K  > globalcities.ps

psxy .....                           -O -K >> globalcities.ps

pstext ......                        -O -K >> globalcities.ps

pstext .......                          -O >> globalcities.ps
```

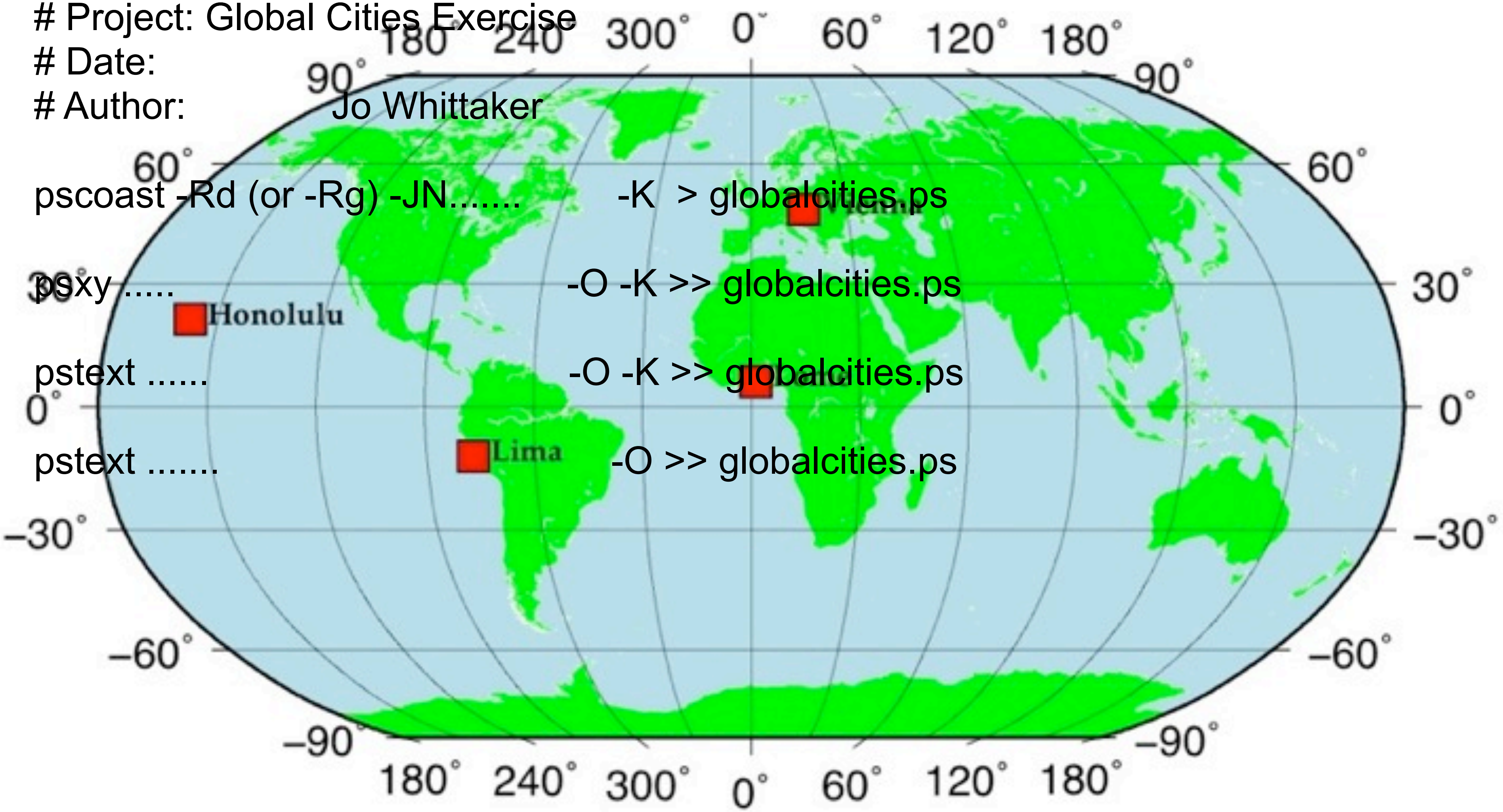


Figure 1: Global map (Robinson Projection), with continents shown in green and oceans shown in light blue. Locations of four cities, Vienna, Honolulu, Lome and Lima are shown as red squares.

```
Honolulu, Lome and Lima are shown as @;red;red@;; squares.
end
```

ksh variables

- To assign variables, use
  - `var=value`
  - `var` must begin with a letter, may contain numbers and underscores, but not $, &.
    - e.g. `size2_big=14`
    - `e.g. inputfile=quakes.d`
- To access the variable, use $var
  - `echo $var`
- You should begin to use variables in your script
  - `e.g. psfile=ex20.ps`

UNIX tools

- Greatly enhances our scripting capability
- Some useful tools

awk	Text processing language
sed	Text substitution stream editor
grep	Pattern matching
wc	Word and record counting
head and tail	List first and last lines from files
sort	General-purpose sorter

awk

- Most important processing tool
- It is often used to reformat data files into a format that GMT can read
- Also used to manipulate data
- Automatically works on each record
- **`$1`**, **`$2`**, etc., are the data fields
- **`$0`** is the entire record
- **`NR`** is record number
- **`NF`** is the number of current fields

Examples of awk commands

- To print only columns 1 and 2 from a file:

```
awk ‘{print $1, $2}’ $infile > $outfile
```

- To print columns 1 and 2 in reverse order:

```
awk ‘{print $2, $1}’ $infile > $outfile
```

- To print column 1 and divide column 2 by 10

```
awk ‘{print $1, $2/10}’ $infile > $outfile
```

Examples of awk commands

- To print only records which contain “Australia” in the third column:

```
awk ‘{if ($3 == “Australia”) print $1,
$2, $3, $4}’ $infile > $outfile
```

- To print only records which do not contain “Australia” in the third column:

```
awk ‘{if ($3 != “Australia”) print $1,
$2, $3, $4}’ $infile > $outfile
```

Smithsonian Volcano Catalog

- We will be using data from the Smithsonian Volcano Catalog (http://www.volcano.si.edu/gvp/world/summinfo.cfm)

- Located in your tutorial directory (volcanoes.dat)
- Open the file and see what it contains

Exercise: Volcanoes

Plotting the GVP data

We will write a script that

1. Makes a global Hammer projection map with blue oceans and light brown continents
2. Plots volcanoes as red triangles

1. Plotting the background

- Basic pscoast job, no further instructions needed!

```
#!/bin/ksh
# Project: Global Volcanoes Exercise
# Date:
# Author:            Jo Whittaker

psfile=globalvolcanoes.ps

pscoast -Rd (or -Rg) -JH.......          -K  > $psfile
```

Reformatting the data

- Lucky for you, we have already formatted the volcano dataset – volcanoes_processed.dat

- Make your plot!

```
#!/bin/ksh
# Project: Global Volcanoes Exercise
# Date:
# Author:              Jo Whittaker

psfile=globalvolcanoes.ps
volocanoes= volcanoes_processed.dat
pscoast -Rd (or -Rg) -JH.......          -K  > $psfile
pscoast -Rd -JH15 -Ba60g30/a30g30 -K -V -Dc -Glightbrown -Slightblue -P -Wwhite
> $psfile.
psxy......                               -O >> $psfile

psxy volcanoes_processed.dat -Rd -J -O -V -St0.1 -Gred  >> $psfile
```

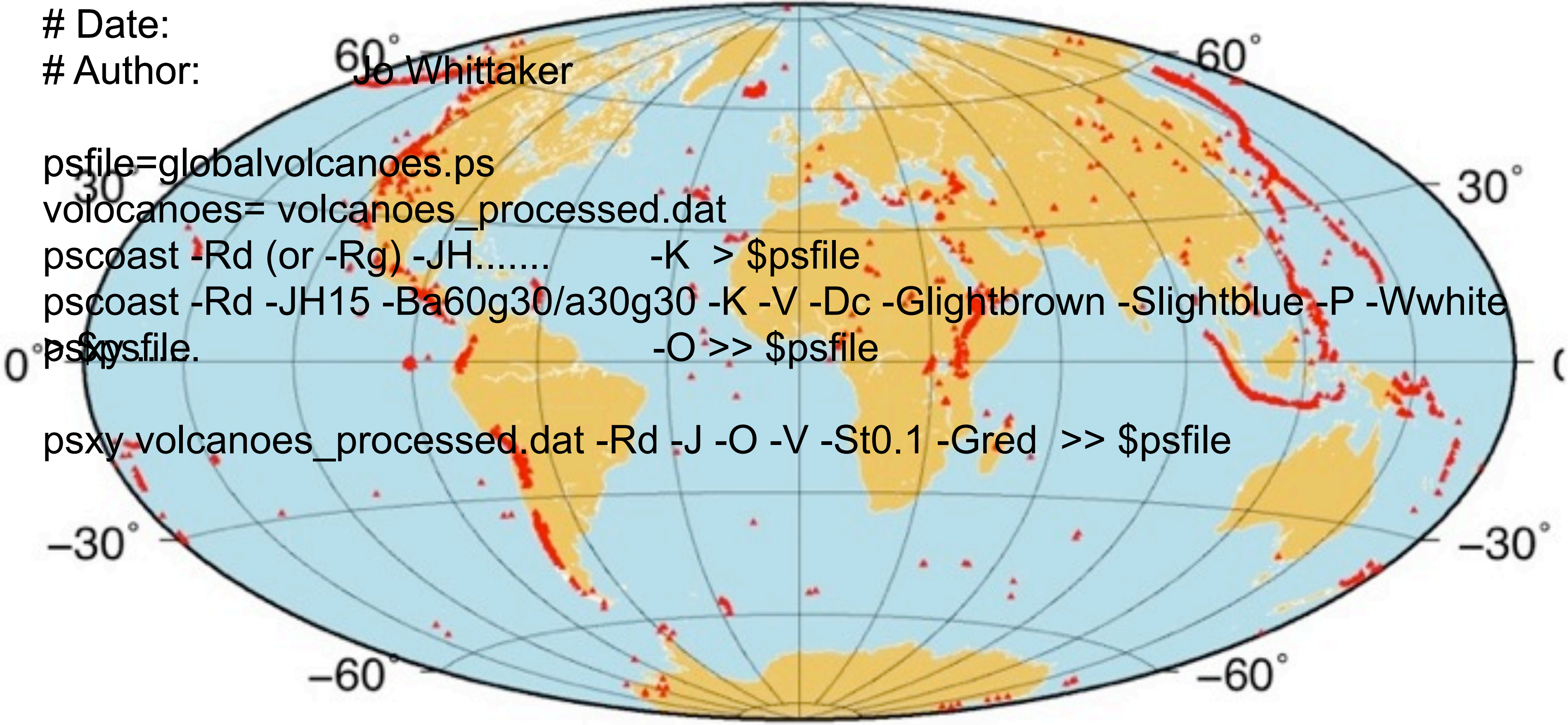
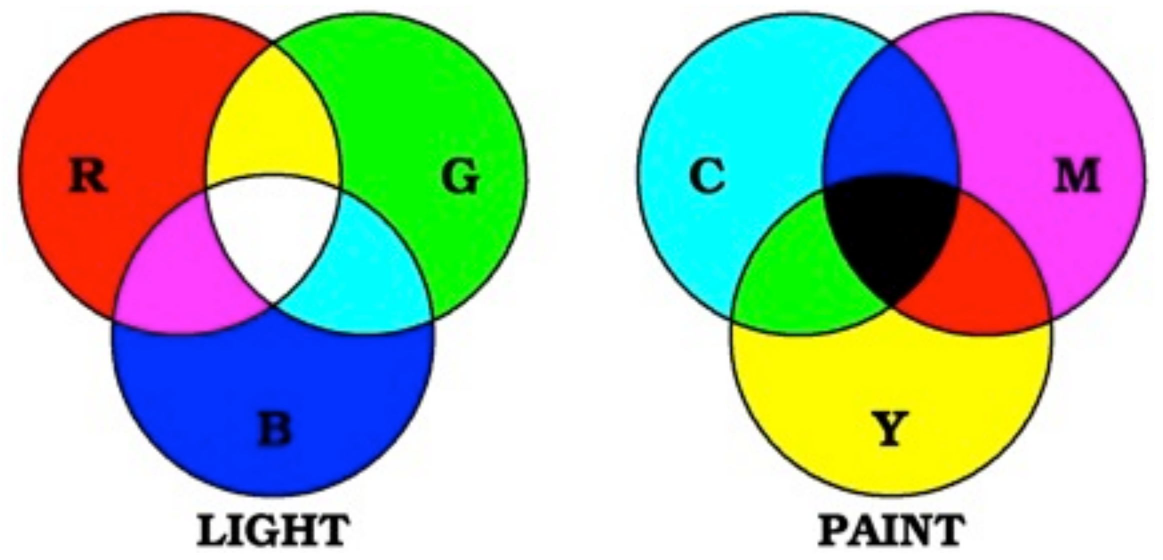


Figure 1: Global map (Hammer Projection), with continents shown in lightbrown and oceans shown in light blue. Locations of volcanoes are shown as red triangles.

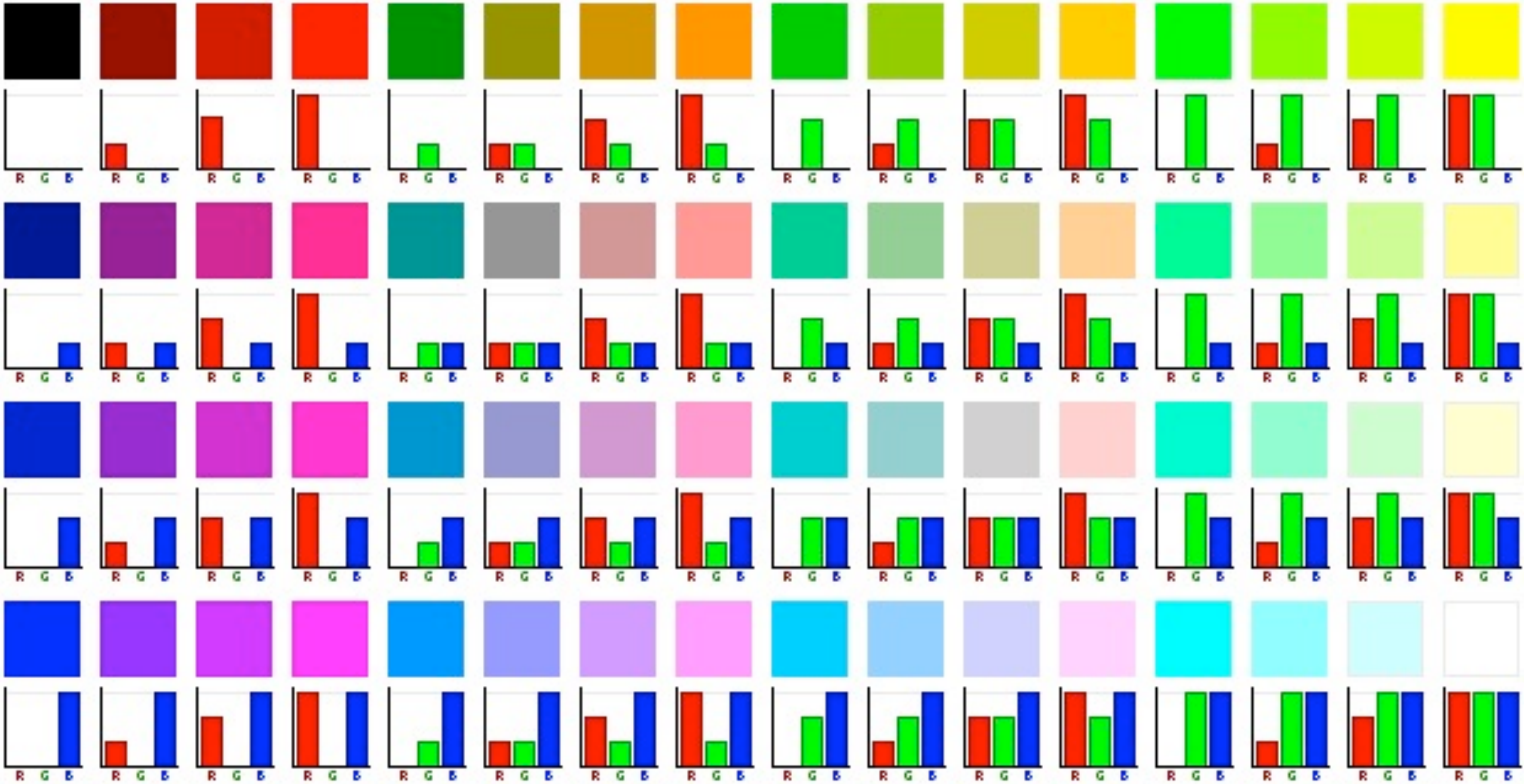
How does color work (again)?



- Computer monitors mix light to make colors
  - RGB is always end-product
- Printers mix paint to make colors
  - Black (K) is used as 4th paint
  - CMY are reduced given the amount of K present

RGB

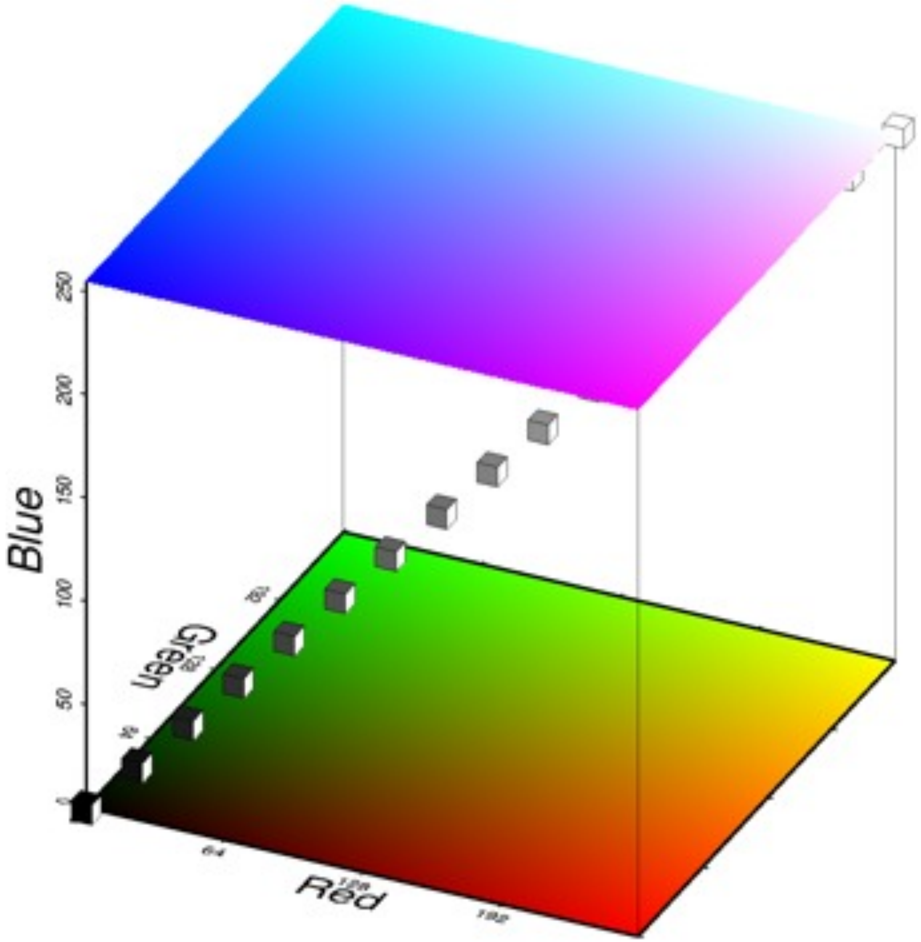
- Different intensities of Red, Green and Blue combine to form different colours



RGB

- Color names: Give standard X11 names such as red, green, violet, pink, lemonchiffon.
- RGB system: Give r/g/b where each integer indicates intensity of light from 0 to 255. If r = g = b we have gray and only r needs to be specified.
- E.g. red = 255/000/000
- E.g. yellow = 255/255/000
- E.g. pink = 200/000/080

The Cartesian RGB System



- Orthogonal coordinate system
- Diagonal R = G = B is called the GREY axis
- The 8 cube corners are primary colors:
  - Red, Green, Blue, White, Cyan, Magenta, Yellow, BlacK

The cpt file

- Created by makecpt, awk, or by hand
- Contains one or more records of format

  z_0 colour$_0$ z_1 colour$_1$ [**U**|**L**|**B**]

- z_0–z_1 is the “z” range of this particular slice
- color$_0$ and color$_1$ specify color for this range
  - Color is constant if color$_0$ = color$_1$
  - Otherwise, it will vary linearly from z_0 to z_1
- Give colors in Gray, RGB, HSV, CMYK or name
  - components must be separated by space or tab
- **U**, **L**, **B** flag indicates we want to annotate the Upper, Lower, or Both ends of the slice

Plot colorbars with psscale

- psscale reads a cpt file and plots a color bar
  - If no file is given it reads from stdin
- May apply artificial illumination to color
- Positioning of scale is done in plot units

Option	Effect
–**C**cptfile	Cpt file to use [stdin]
–**D**x/y/len/width[**h**]	Placement of bar
–**I**max_intens	Add illumination effect

Exercise: plotting colour with psscale

- Pick a preset color table (type makecpt to find out the colour palettes available

- Make a discrete colour palette using makecpt

```
makecpt -C$cpt -T0/1/0.1 > ${cpt}_discrete.cpt
```

- Make a continuous colour palette

```
makecpt -C$cpt -T0/1/0.1 -Z > $cpt_continu.cpt
```

- Open both files and see the difference

Exercise: plotting colour with psscale

- Start a new Script and plot the discrete color palette using psscale

```
psscale -C$cpt_discrete.cpt -P -D8/4/10/1h -
   B:"Discrete $cpt Colour Palette": > $psfile
```

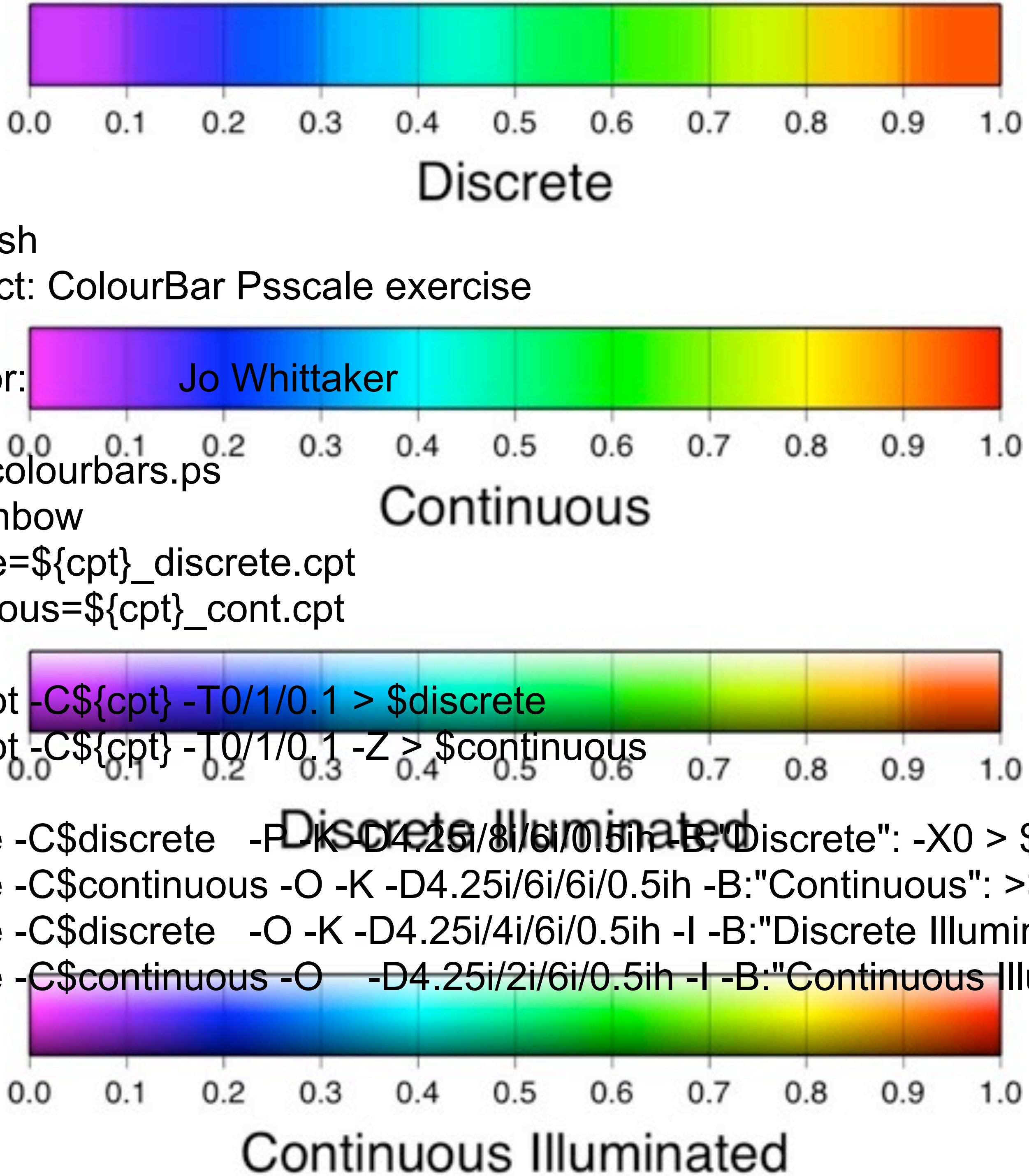
- Plot the continuous colour palette above it on the same page. (Be careful with -K, -O and > vs >>)

```
psscale -C$cpt_discrete.cpt -P -D8/4/10/1h -
   B:"Discrete $cpt Colour Palette": -K >
   $psfile
psscale -C$cpt_continu.cpt -D8/8/10/1h -O -
   B:"Continuous $cpt Colour Palette": >>
   $psfile
```

Exercise: plotting colour with psscale

Make a script that creates and plots 4 vertical colourbars on the same page.

- 2 Discrete – with and without illumination
- 2 Continuous – with and without illumination



```
#!/bin/ksh
# Project: ColourBar Psscale exercise
# Date:
# Author:        Jo Whittaker

psfile=colourbars.ps
cpt=rainbow
discrete=${cpt}_discrete.cpt
continuous=${cpt}_cont.cpt

makecpt -C${cpt} -T0/1/0.1 > $discrete
makecpt -C${cpt} -T0/1/0.1 -Z > $continuous

psscale -C$discrete   -P -K -D4.25i/8i/6i/0.5ih -B:"Discrete": -X0 > $psfile
psscale -C$continuous -O -K -D4.25i/6i/6i/0.5ih -B:"Continuous": >> $psfile
psscale -C$discrete   -O -K -D4.25i/4i/6i/0.5ih -I -B:"Discrete Illuminated": >> $psfile
psscale -C$continuous -O    -D4.25i/2i/6i/0.5ih -I -B:"Continuous Illuminated": >> $psfile
```

Exercise: Plotting Pacific Seamounts

You have a file in your tutorial directory called:

Pacific_ages.d

It has ages of seamounts, islands, and plateaus in the Pacific. Look at this file using Notepad ++.

Exercise: Pacific Seamounts

Write a script that makes a Mercator projected map of Hawaii-Emperor area (west=150, east =210, south=0, north=60)

- Extract records for Hawaii-Emperor only [already done - use seamount_pac.d]
- Plot dated seamounts/islands using color-coded triangles [psxy -C]
- Create cptfile based on ages of seamounts [makecpt]
- Plot the number of samples in the title [wc]
- Place your color table beneath the map [psscale]

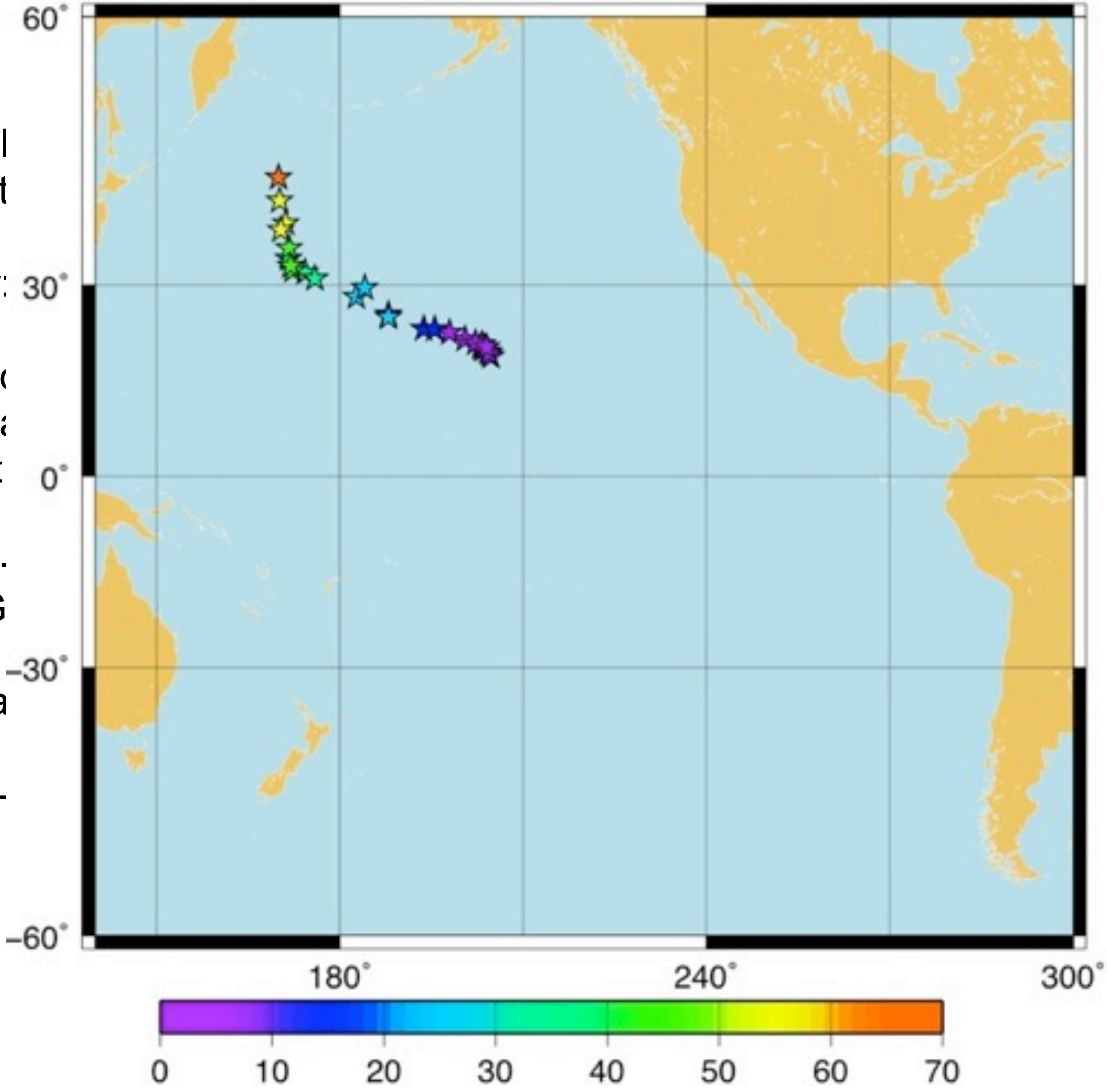
```
#!/bin/ksl
# Project
# Date:
# Author:

psfile=gl
cpt=volca
makecpt

pscoast -                                  = 28":neSW -K -
V -Dc -G

psxy sea

psscale -
```



```
#!/bin/ksh
# Project: Global Volcanoes Exercise
# Date:
# Author:		Jo Whittaker

psfile=globalvolcanoes.ps
cpt=volcanoes.cpt
makecpt -Crainbow -T0/70/10 > $cpt

pscoast -R140/300/-60/60 -JM15 -Ba60g30/a30g30:."No. of Volcanoes = 28":neSW -K -V -Dc -Glightbrown -Slightblue -P -Wwhite > $psfile

psxy seamount_pac.d -R -J -O -K -V -Sa0.4 -Wthin  -C$cpt >> $psfile

psscale -C$cpt -O -K -V -D7/-1/12/0.5h >> $psfile
```