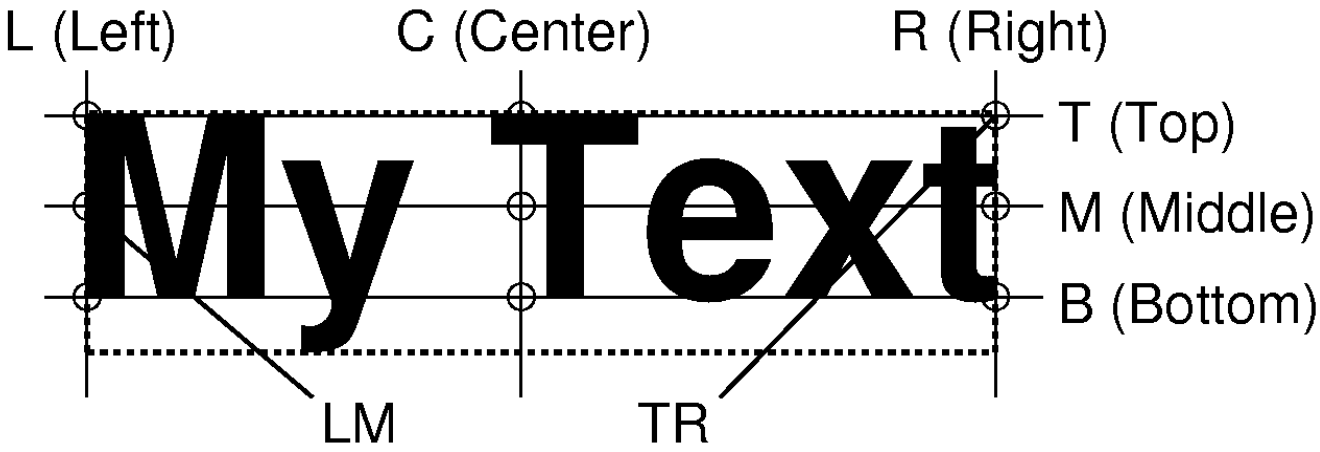
GMT4 pstext text line input and setup

x y size angle font justify text

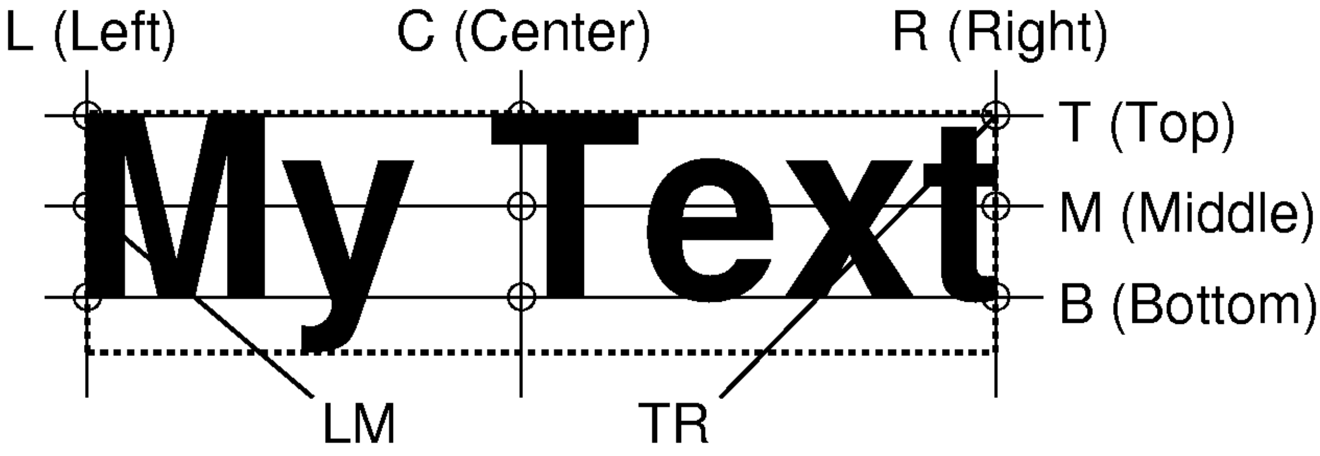
- x y is map coordinates
- size is font size in points
- angle is inclination of text baseline w.r.t. horizontal
- font is font name or number (see **–L**)
- justify relates (x, y) to a point on the text string
- text



GMT5 pstext text line input and setup

x y text

- The following plotting info is captured by the -F flag
  - size is font size in points (-F+f16p)
  - angle is inclination of text baseline w.r.t. horizontal (-F+a45)
  - font is font name or number (-F+f16p,Courier)
  - justify relates (x, y) to a point on the text string (-F+jMC)
  - Keep all together: -F+f16p,Helvetica,blue+a45+jMC

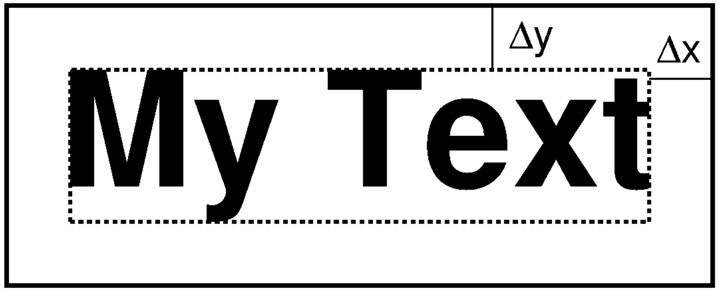


Plotting text lines and paragraphs

- 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**)

Different in GMT5!!!!



Plotting text – pstext (GMT4 vs GMT5)

- The GMT5's pstext is very different from the GMT4's pstext
- GMT5 is much simpler as the all the text plotting info (size, colour...) is given in a –F flag
- Note the use of echo and a pipe below:

```
echo “120 20 16 45 Courier MC hello” | pstext –Rg -JW120/20c –Gmagenta –V >
gmt4_pstext.ps #GMT4
```

...same as...

```
echo “120 20 hello” | gmt pstext –Rg -JW120/20c –
F+a45+f16p,Courier,magenta+jMC –V > gmt5_pstext.ps #GMT5
```

Exercise 14: Plot your name

- Create a simple linear plot and write your name using a font size of 10pt, font type of Palatino-Bold and left justified
- 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

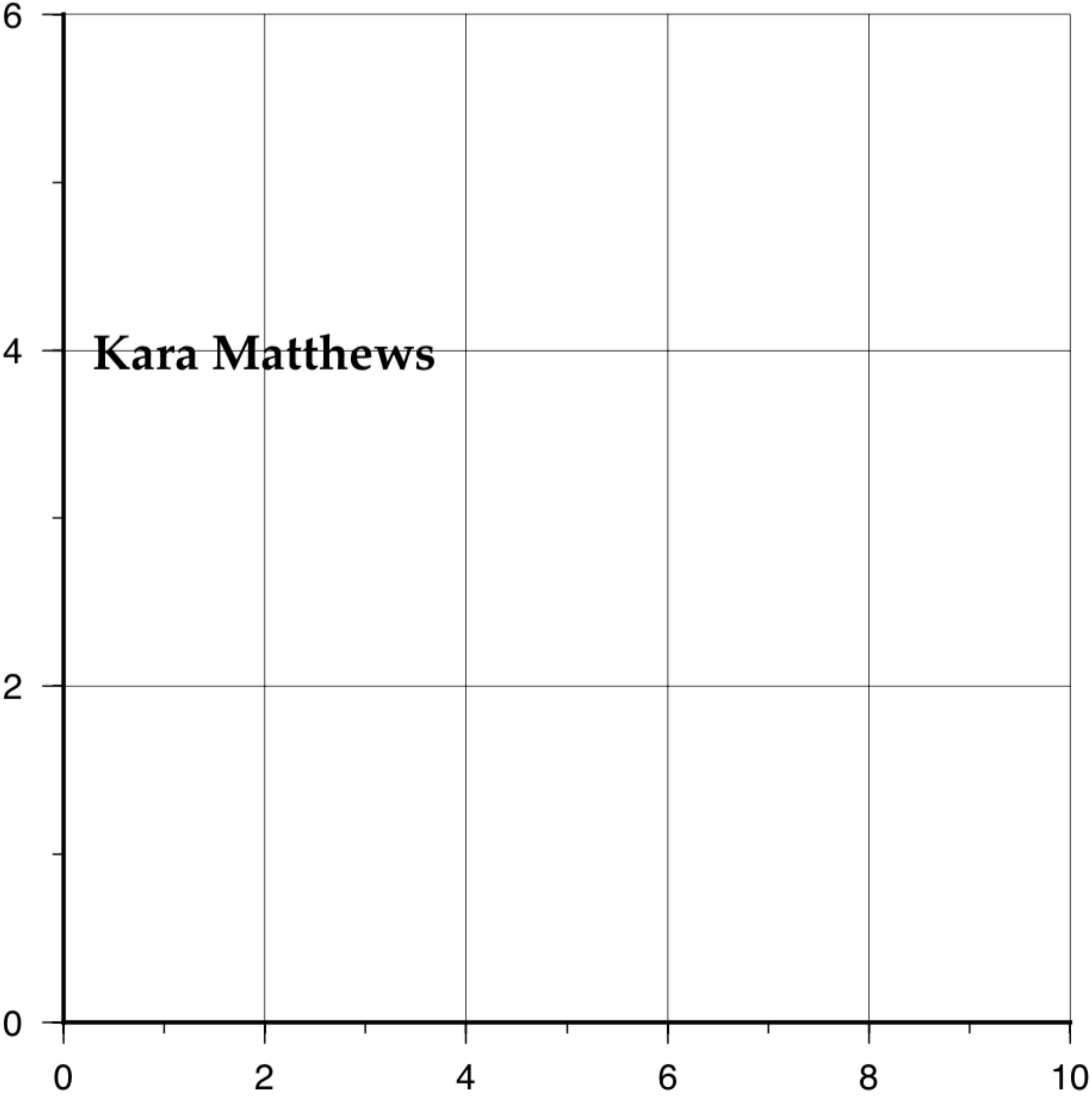
```
2 4 14     0      31    CM     Your Name here
```

or

```
2 4 Your Name here  <GMT5>
```

GMT4

```
pstext MyName.txt -R0/10/0/6 -JX10 -Ba2f1g2WS -P >
ex14.ps
```


6
4
2
0
Kara Matthews
0
2
4
6
8
10

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 > ex14.ps
2 4 14 0 31 CM Your Name here
end
```

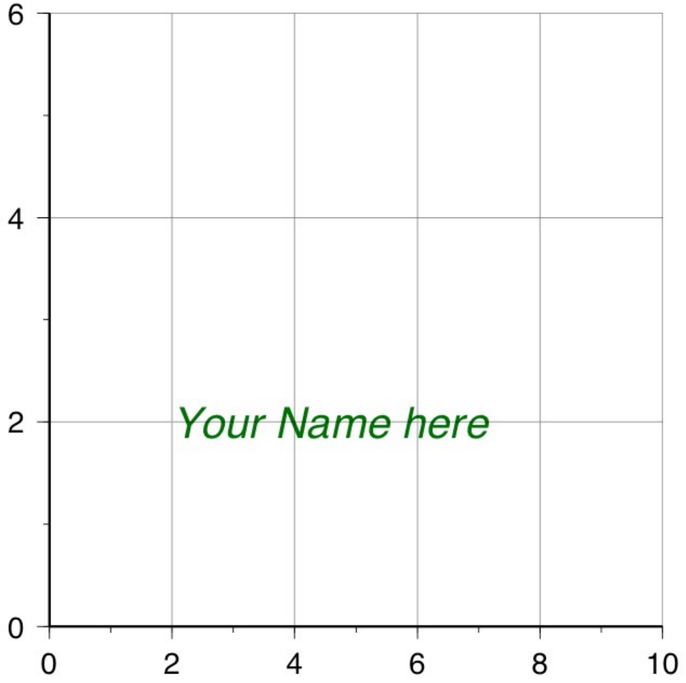
or

```
echo “2 4 14 0 31 CM Your Name here” | pstext -
R0/10/0/6 -JX10 -Ba2f1g2WS -P > ex14.ps
```

Exercise 15: Plot your name cont…

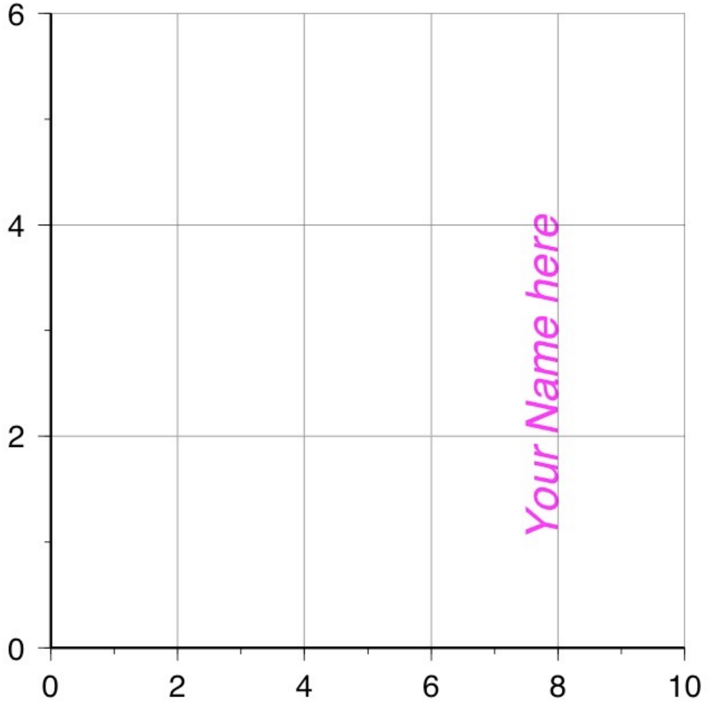
- Plot your name using font size of 20 and a dark green Helvetica-Oblique font

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



```
pstext -R0/10/0/6 -JX10 -Ba2f1g2WS -P -Gdarkgreen  <<end > ex12.ps
2 2 20 0 2 LM Your Name here
end
```

```
pstext -R0/10/0/6 -JX10 -Ba2f1g2WS -P -Gmagenta <<end > ex12.ps
8 1 20 90 2 1 Your Name here
end
```



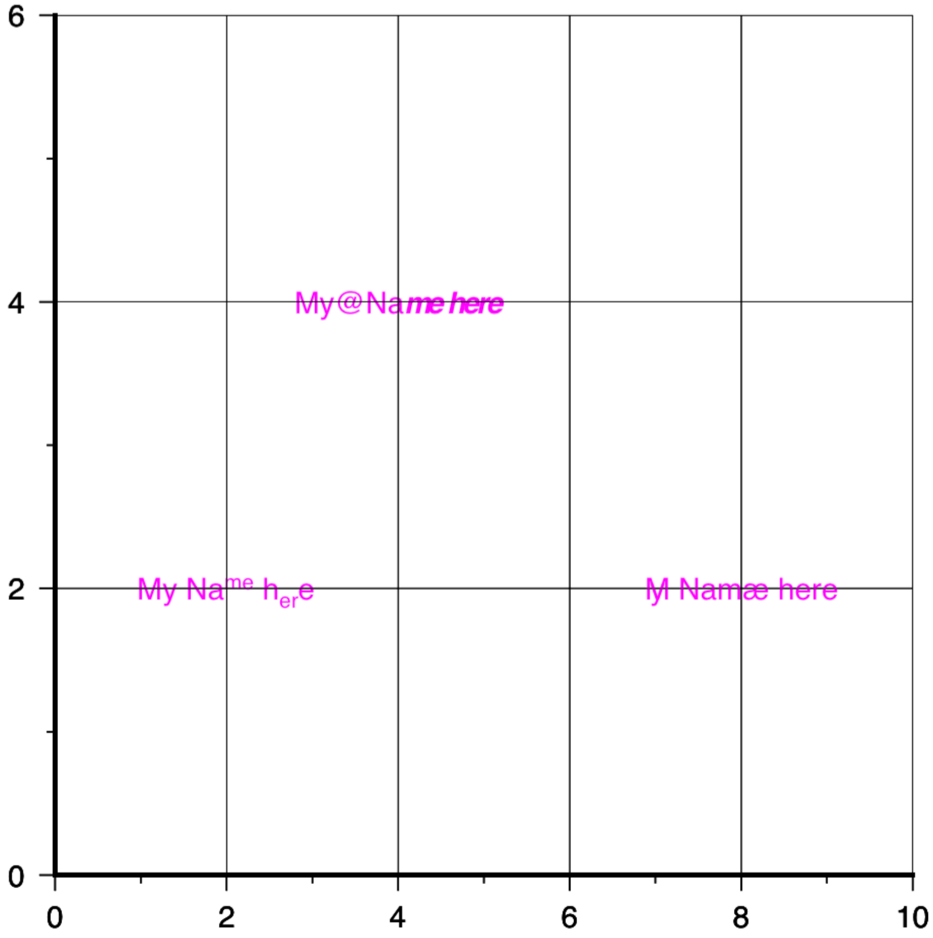
GMT text escape sequences

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

Example 16

Try plotting the following “My Name here” text:

```
My@@Na@%24%me here
My Na@+me@+ h@-er@-e
@!My Nam@e here
```



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

Exercise 17 pstext in Paragraph mode

Create a file called MyParagraph.txt that looks roughly like: (you don’t have to copy all the text word for word)

remember to move to -F if using GMT5

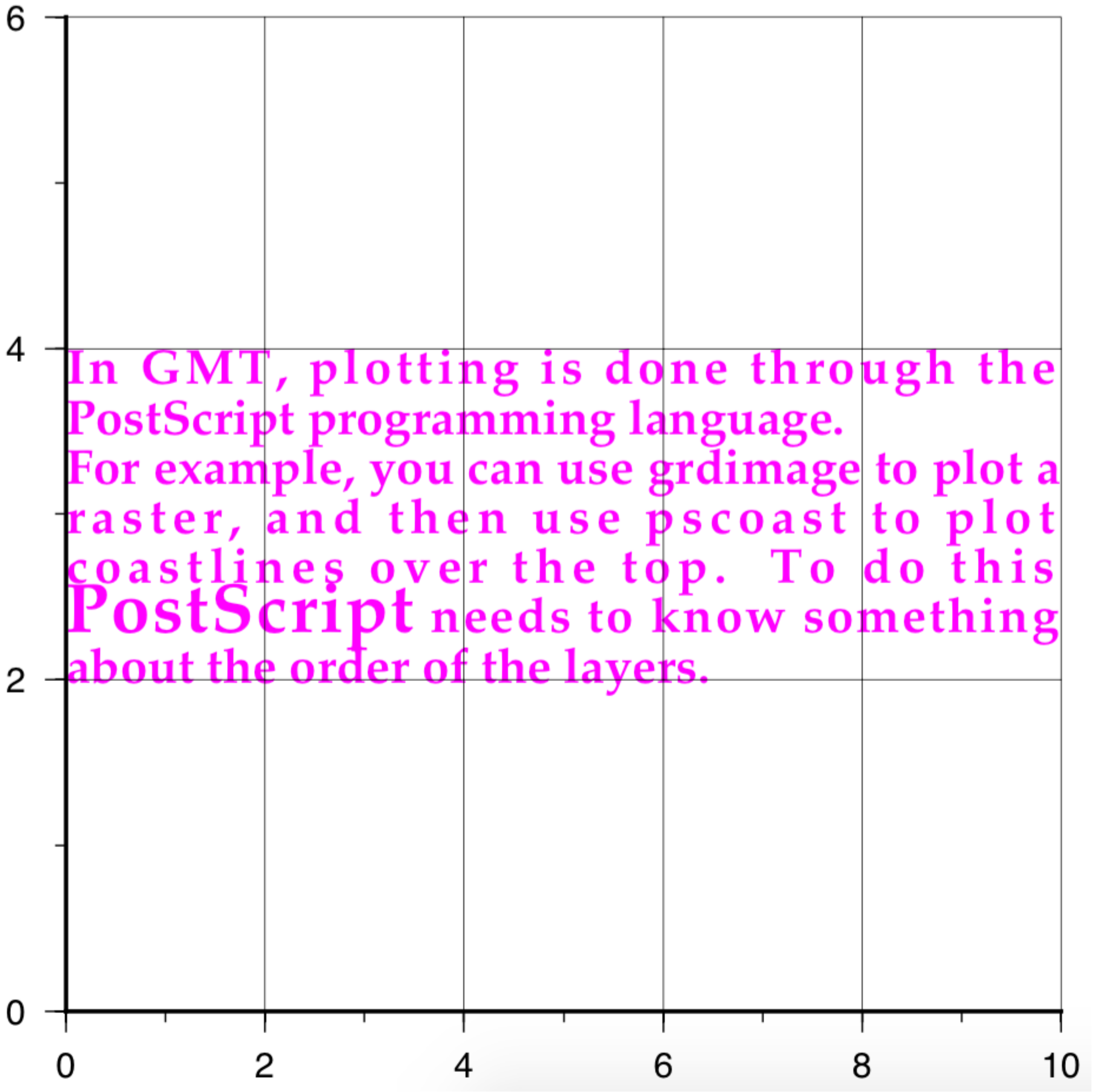
```
> 0 4 14 0 31 LT 0.5 10 j
In GMT, plotting is done through the PostScript programming
language.


For example, you can use grdimage to plot a raster, and then
use pscoast to plot coastlines over the top. To do this @:
20:PostScript@:: needs to know something about the order of
the layers.
```

Command looks like:

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

!!!remember to add a -F if using GMT5!!!

Exercise 17 pstext in Paragraph mode



Exercise 18: 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)
- 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 consult the manual for pstext

```
#!/bin/bash
# Project: Global Cities Exercise
# Date:
# Author:

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

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

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

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

!!!Adjust red text if using GMT5 and remember to add a -F flag!!!

```
#!/bin/bash
# Project: Global Cities Exercise
# Date:
# Author:

pscoast -Rd -JN15 -Ba60g30/a30g30 -K -V -Dc -Ggreen -Slightblue -P -Wwhite >
globalcities.ps

psxy MyCities.txt -Rd -J -O -K -V -Ss0.5 -Wthin -Gred  >> globalcities.ps

pstext -R -J -Gblack -O -K -V -D0.2 << end >> globalcities.ps
16.36   48.21 10 0 31 LT Vienna
-157.86 21.31 10 0 31 LT Honolulu
1.22     6.11 10 0 31 LT Lome
-77.04 -12.05 10 0 31 LT Lima
end

pstext -R -J -Gblack -O -M -V -N -X-8 -Y-5 << end >> globalcities.ps
> 0 0 10 0 31 LT 0.5 15 j
@_Figure 1:@_ Global map (Robinson Projection), with continents shown in @;
0/255/0;green@;;
 and oceans shown in @;lightblue;light blue@;;. Locations of four cities, Vienna,
Honolulu, Lome and Lima are shown as @;red;red@;; squares.
end
```

```
#!/bin/bash
# Project: Global Cities Exercise
# Date:
# Author:

pscoast -Rd -JN15 -Ba60g30/a30g30 -K -V -Dc -Ggreen -Slightblue -P -
Wwhite > globalcities.ps

psxy MyCities.txt -Rd -J -O -K -V -Ss0.5 -Wthin -Gred  >> globalcities.ps

pstext -R -J -Gblack -O -K -V -D0.2 << end >> globalcities.ps
16.36   48.21 10 0 31 LT Vienna
-157.86 21.31 10 0 31 LT Honolulu
1.22     6.11 10 0 31 LT Lome
-77.04 -12.05 10 0 31 LT Lima
end

pstext -R -J -Gblack -O -M -V -N -X-8 -Y-5 << end >> globalcities.ps
> 0 0 10 0 31 LT 0.5 15 j
@_Figure 1:@_ Global map (Robinson Projection), with continents shown in
@;0/255/0;green@;;
 and oceans shown in @;lightblue;light blue@;;. Locations of four cities,
Vienna,
Honolulu, Lome and Lima are shown as @;red;red@;; squares.
end
```

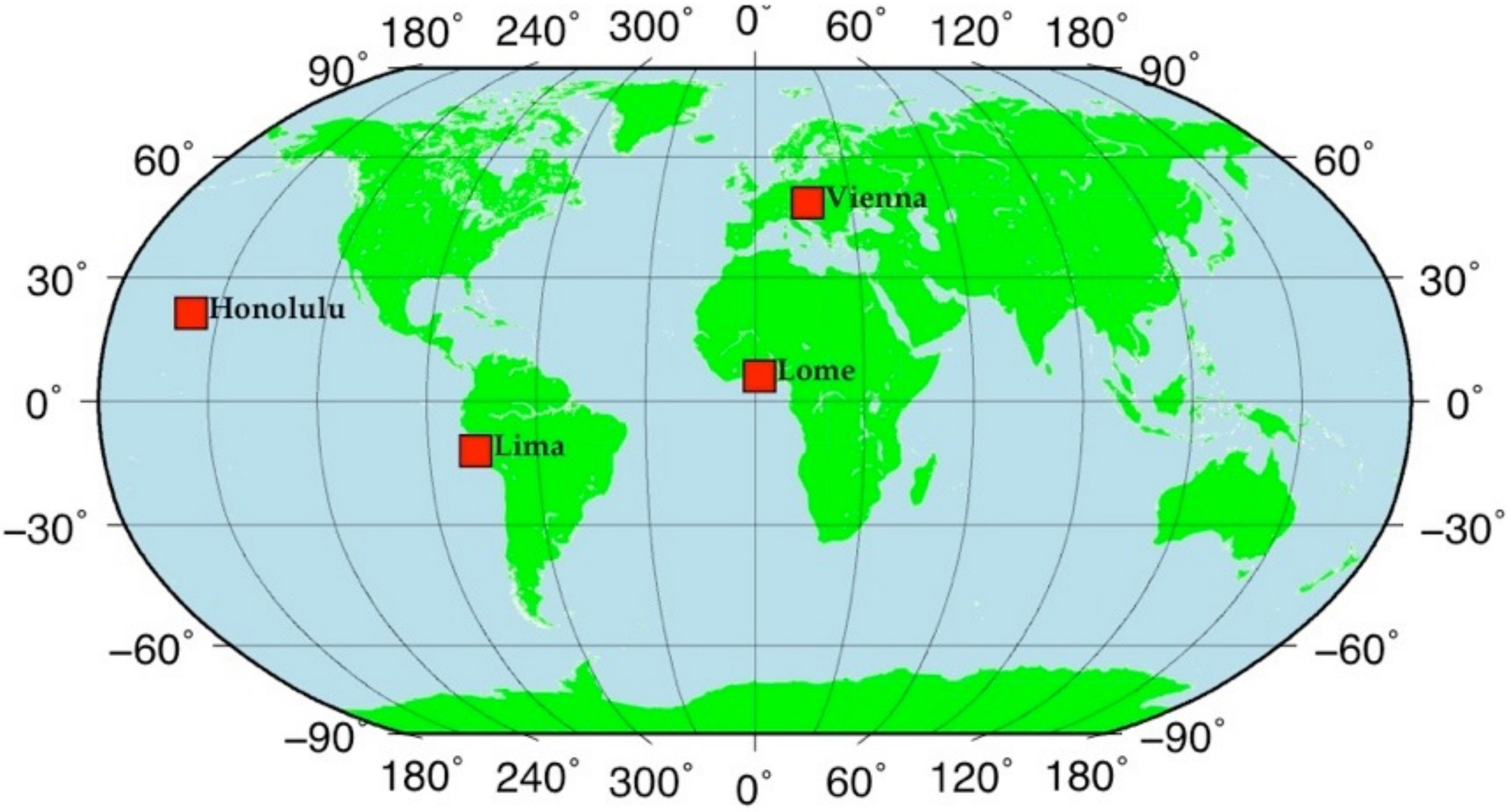


Figure 1: Global map (Robinson Projection), with continents shown in green and oceans shown in light blue. Locations of four cities, Vienna, Honolulu, Lome and Lima are shown as red squares.

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
```

```
awk '{print $1, $2, "some text", $2/10}'
$infile > $outfile
```

Examples of awk commands

- Print to stdout only records which contain “Sydney” in the third column:

```
awk ‘{if ($3 == “Sydney”) print $1, $2,
$3}’ wcity_major.dat
```

- Print to stdout only records which do not contain “Sydney” in the third column:

```
awk ‘{if ($3 != “Sydney”) print $1, $2,
$3}’ wcity_major.dat
```

Smithsonian Volcano Catalog

- We will be using data from the Smithsonian Volcano Catalog (http://www.volcano.si.edu/gvp/world/summinfo.cfm)
- Location: Data/Day2/volcanoes.dat
- Open the file and see what it contains
  - open in a text editor
  - type “head volcanoes.dat” in the terminal

Exercise 19: 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
3. Plot a figure caption

1. Plotting the background

- Basic pscoast job, no further instructions needed!

```
#!/bin/bash
# Project: Global Volcanoes Exercise
# Date:
# Author:

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/bash
 # Project: Global Volcanoes Exercise
 # Date:
 # Author:
 psfile=globalvolcanoes.ps

pscoast -Rd -JH15 -Ba60g30/a30g30 -K -V -Dc -Glightbrown -Slightblue -P -Wwhite > $psfile

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

pstext -R -J -Gblack -O -M -V -N -X-8 -Y-5 << end >> $psfile
> 0 0 10 0 31 LT 0.5 15 j
@_Figure 1:@_ Global map (Hammer Projection), with continents shown in @;lightbrown;lightbrown@;;
 and oceans shown in @;lightblue;light blue@;;. Locations of volcanoes
 are shown as @;red;red@;; triangles.
end
```

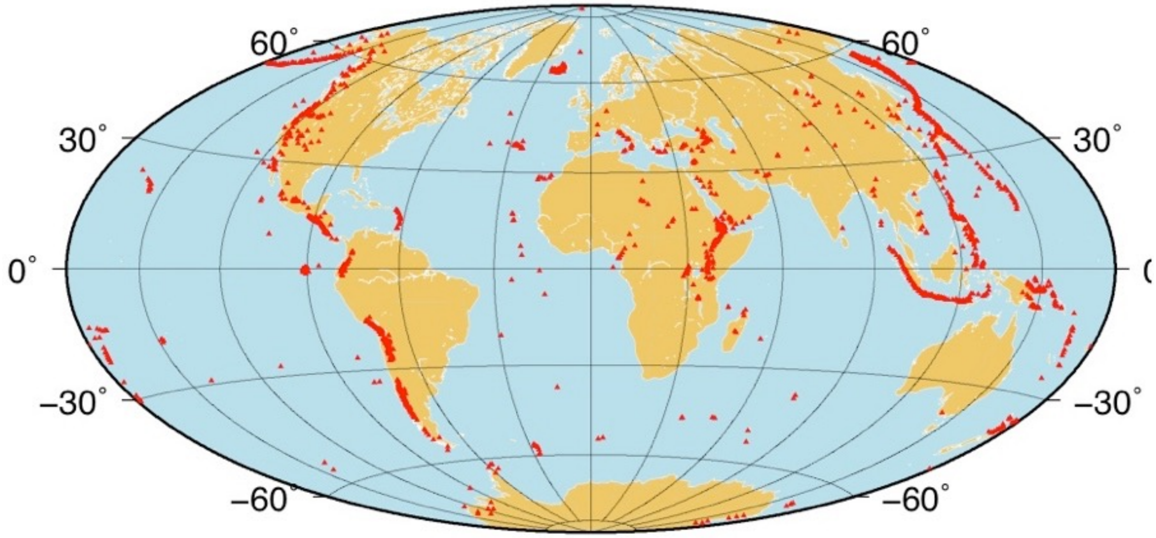
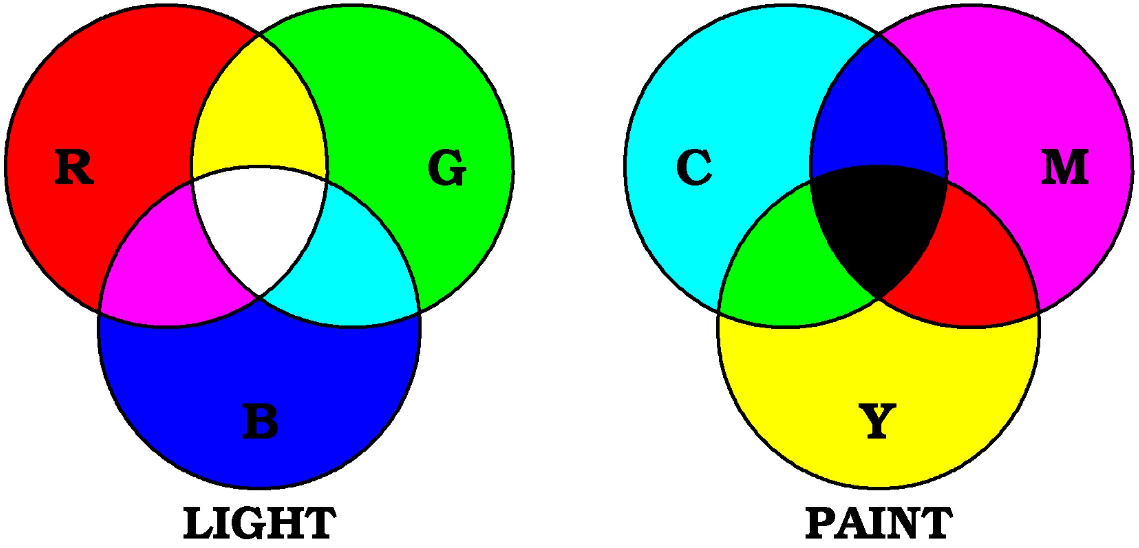


Figure 1: Global map (Hammer Projection), with continents shown in lightbrown and oceans shown in light blue. Locations of volcanoes are shown as red triangles.

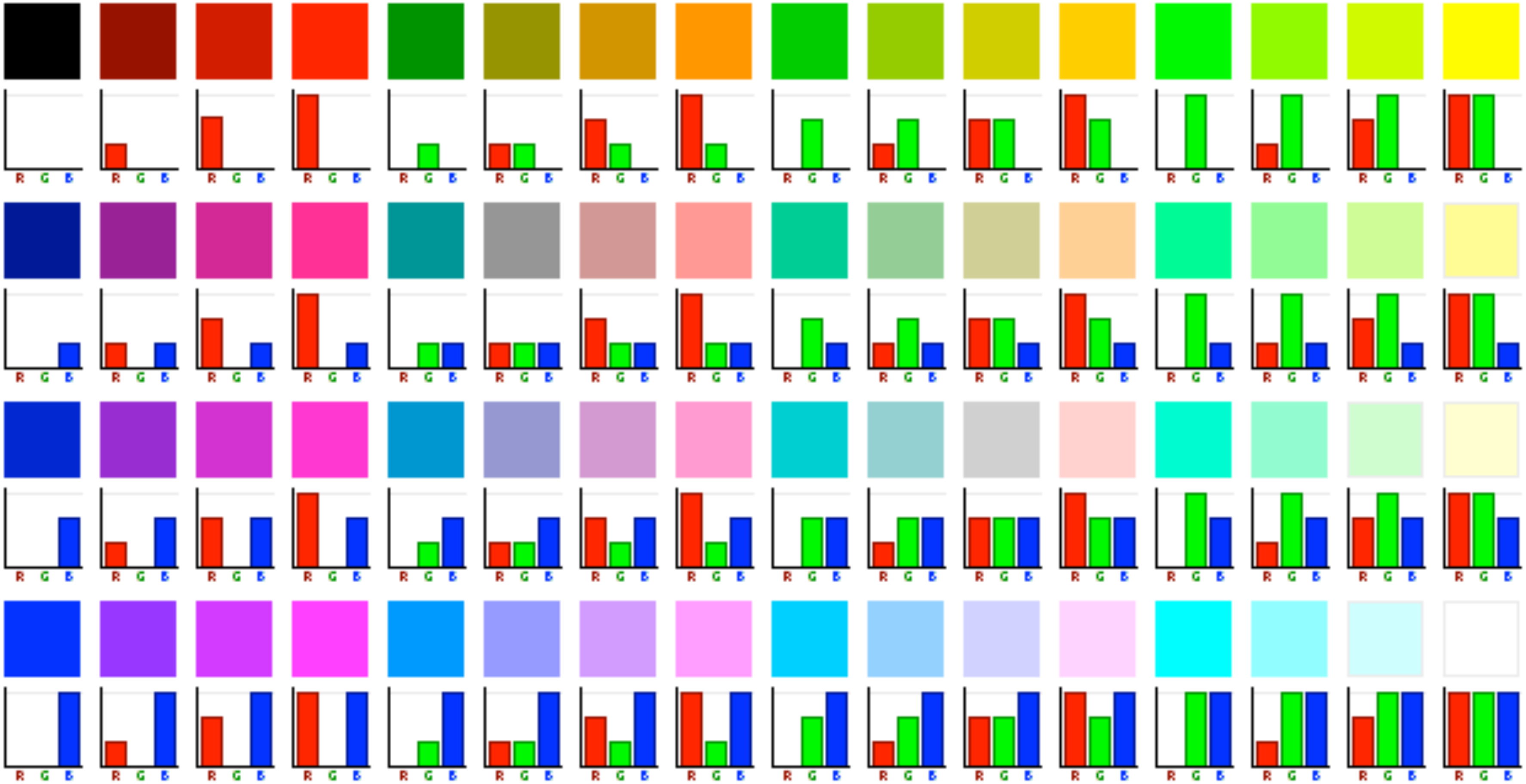
How does color work (again)?



- 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

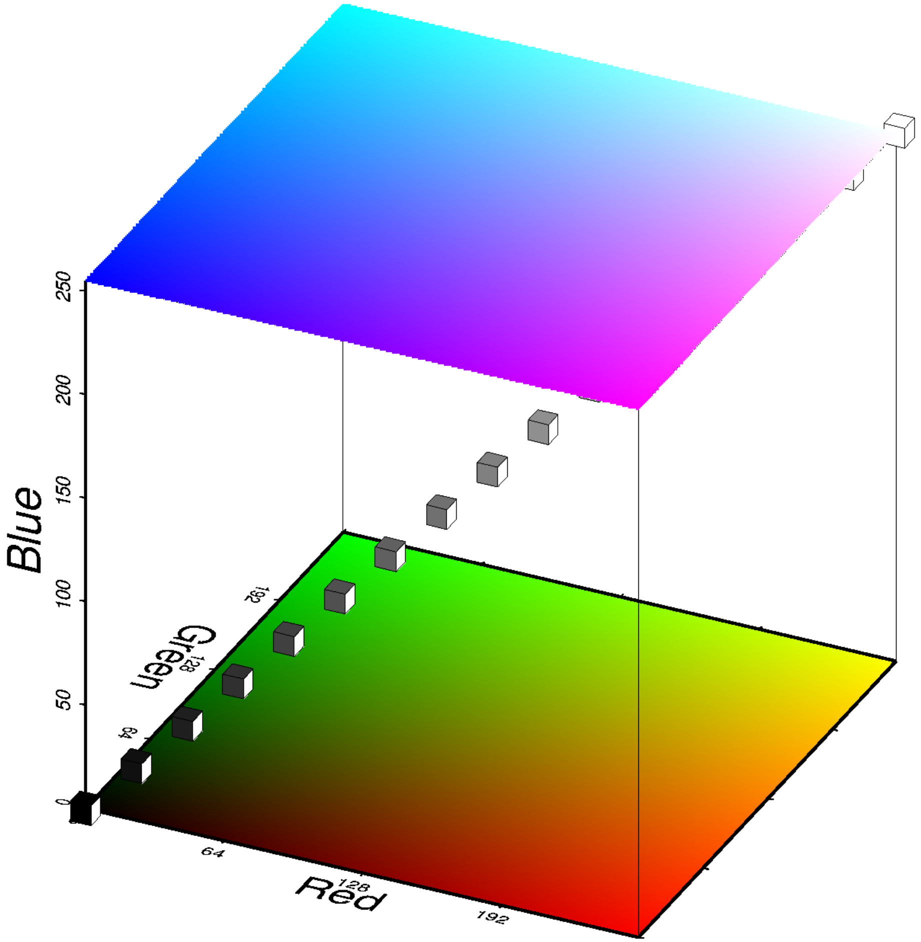
- Different intensities of Red, Green and Blue combine to form different colours



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]
–Dx/y/len/width[**h**]	Placement of bar
–Imax_intens	Add illumination effect

Exercise 20: plotting colour with psscale

- Start a new script called Ex20.sh
- type makecpt to find out the colour palettes available... we will use wysiwyg (what you see is what you get)
- When you're finished open your cpts and compare
- Make a discrete colour palette using makecpt

```
cpt=wysiwyg
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
```

Exercise 20: 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 >>)

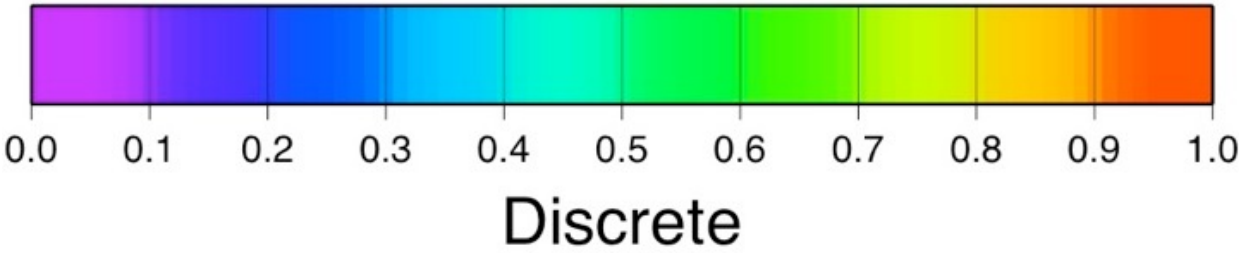
```
psscale -C${cpt}_discrete.cpt -P -D8/4/10/1h
  -B:″Discrete $cpt Colour Palette″: -K >
  $psfile
psscale -C${cpt}_continu.cpt -D8/8/10/1h -O -
  B:″Continuous $cpt Colour Palette″: >>
  $psfile
```

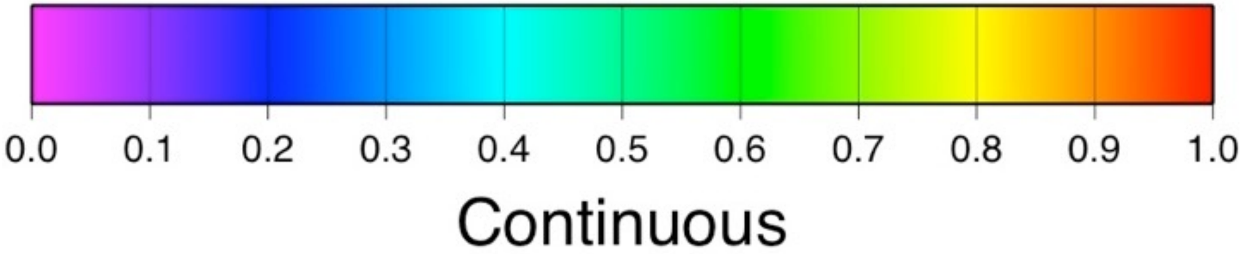
Exercise 21: plotting colour with psscale

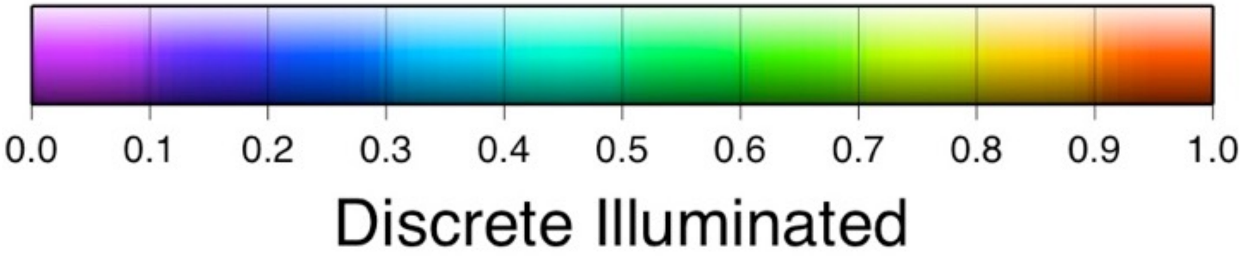
Make a script that creates and plots 4 colourbars on the same page.

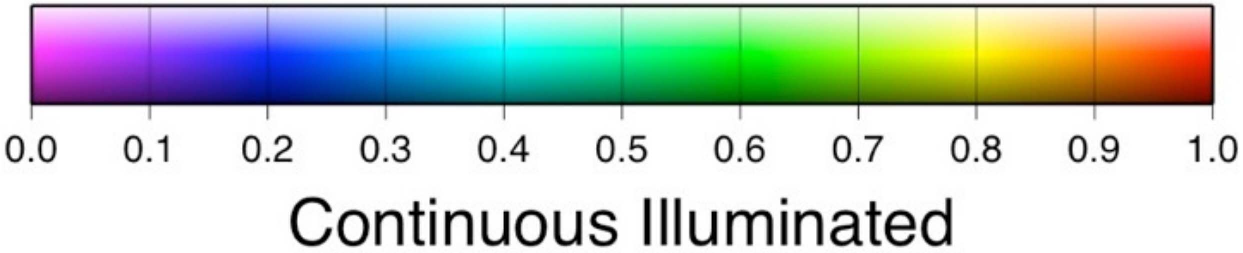
- 2 Discrete – with and without illumination
- 2 Continuous – with and without illumination

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

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

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


0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Discrete


0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Continuous


0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Discrete Illuminated


0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Continuous Illuminated

Exercise 22: 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 a text editor

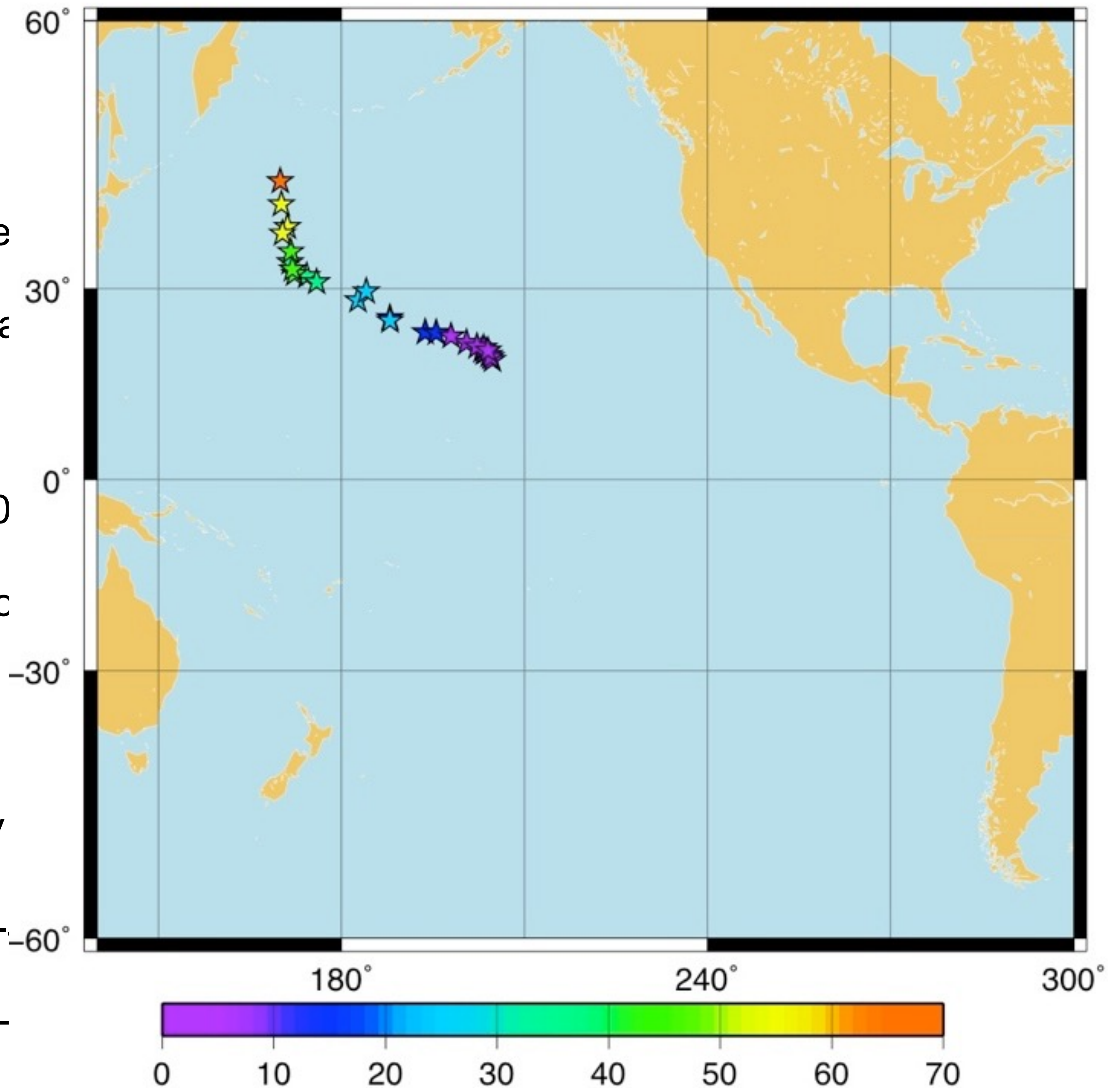
Exercise: Pacific Seamounts

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]

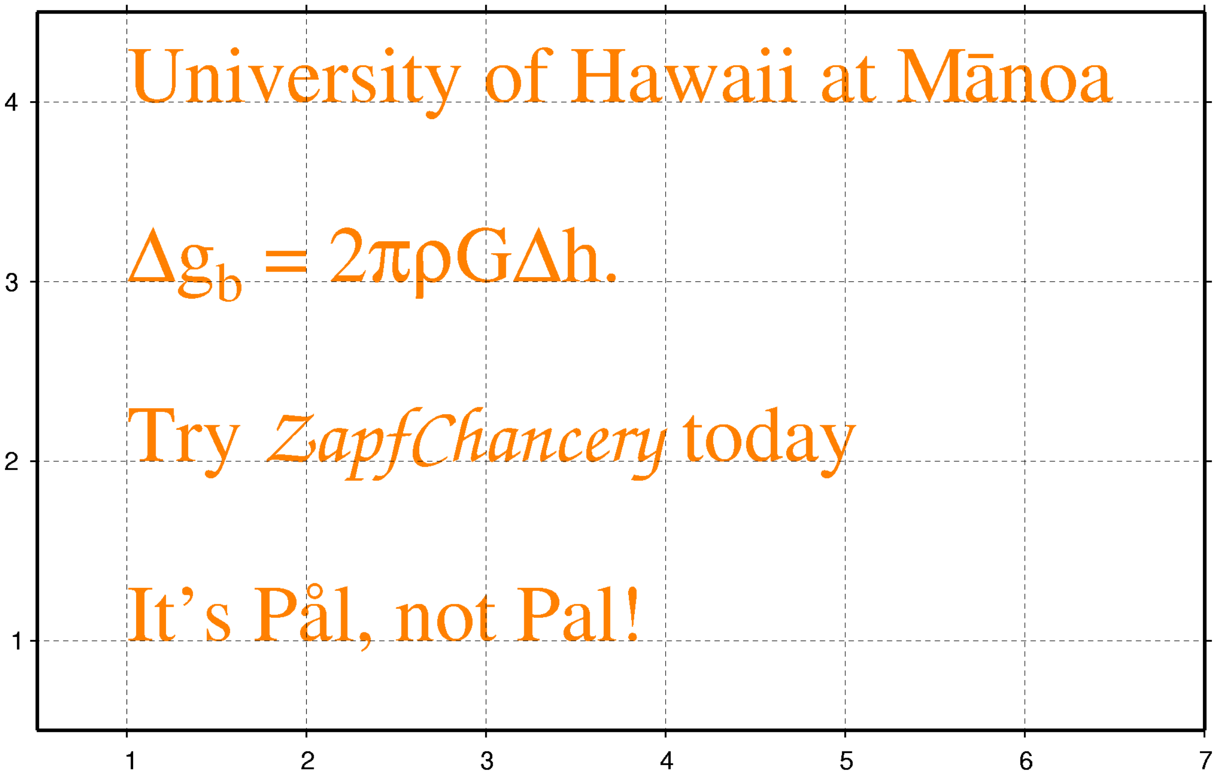
```
#!/bin/ksh
# Project: Global Volcanoes Exercise
# Date:
# Author:               Jo Whittaker

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

volcanoes=$( wc -l  seamount_pac.d | awk ‘{print $1-1}’ )

echo $volcanoes

pscoast -R140/300/-60/60 -JM15 -Ba60g30/a30g30:."No. of Volcanoes = \
${volcanoes}“: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 -V -D7/-1/12/0.5h >> $psfile
```

```
#!/bin/ksh
# Project: Global Volcanoe
# Date:
# Author:             Jo Whitta

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

volcanoes=$( wc -l  seamo

echo $volcanoes

pscoast -R140/300/-60/60
${volcanoes}“:neSW -K -V

psxy seamount_pac.d -R -

psscale -C$cpt -O -V -D7/-
```



Advanced exercise

- Recreate the following plot. Hint: you need to change **CHAR_ENCODING**.


University of Hawaii at Mānoa
$\Delta g_b = 2\pi\rho G\Delta h.$
Try *ZapfChancery* today
It’s Pål, not Pal!