University of California, Los Angeles
Department of Statistics

Statistics C173/C273 **Instructor: Nicolas Christou**

Introduction to Geographical Resources Analysis Support System (GRASS)

General information:

- GRASS Website, **http://grass.itc.it/**.
  GRASS logo:

- GRASS Team: **http://grass.ibiblio.org/community/team.php** .
- GRASS Wiki site **http://grass.osgeo.org/wiki/**.
- GRASS was originally developed for the U.S. Army Construction Engineering Research Laboratories (1982-1995) as a tool for land managing.
- GRASS is a free software used for data management, image processing, graphics production, spatial modelling, and visualization of many types of data.
- It is currently used in Academia, Industry, and Government (NASA, USGS, U.S. Census Bureau, etc.).
- The GRASS database:
  All data and files are saved under the home directory or under a shared network, e.g.

  `~/Users/nicolas/grass`

- Under this directory, the GRASS GIS data are organized by projects stored in subdirectories called **LOCATIONs**. Each location is defined by its coordinates and map projection. Once a new location is created, GRASS internally produces other directories for the handling of the data (not important at this point).
- Further, LOCATIONs are subdivided into map subdirectories called **MAPSETs**. Note: When creating a new location GRASS automatically creates a special mapset called PERMANENT. Each LOCATION can have several MAPSETs.

We can represent the above as follows:

GRASS Database (Home directory)

LOCATION

/A /B /C

MAPSET

/PERMANENT /a1 /a2 /PERMANENT/ b1 /b2 /PERMANENT /c1 /c2

Under a mapset we can save our files that contain maps (either vector or raster with their attributes and other features).

GRASS modules (function classes):
The modules in GRASS are organized by name (based on their function class). The first letter refers to the function class, then it follows by a dot and then by another word that describes the specific task described by the module. For example *d.rast* will display a raster map. Here are the most important modules in GRASS:

First letter	Function class	Type of command
d.∗	display	graphical display
db.∗	database	database management
g.∗	general	general file operations
i.∗	imagery	imagery processing
m.∗	misc	miscellaneous commands
ps.∗	postscript	creation of a Postscript map
r.∗	raster	raster data processing
r3.∗	3D raster	3D raster data processing
v.∗	vector	vector data processing

The general syntax of a GRASS command is similar to UNIX commands.

To get help in GRASS type the module and then "help". For example:

```
d.rast help
```

To open a monitor for map display we type:

```
d.mon x0
```

Note: GRASS can open up to 7 monitors. If more than one monitors are opened, we can select a particular monitor as follows:

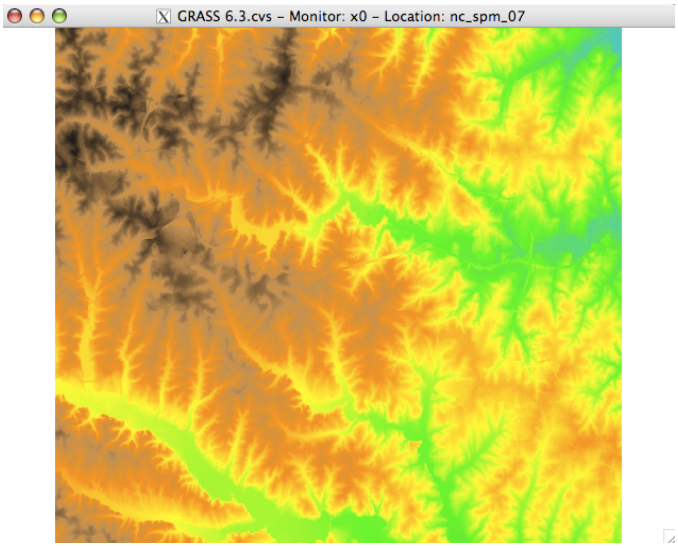
```
d.mon select=x0
```

GRASS manual for version 6.3 and 6.4:

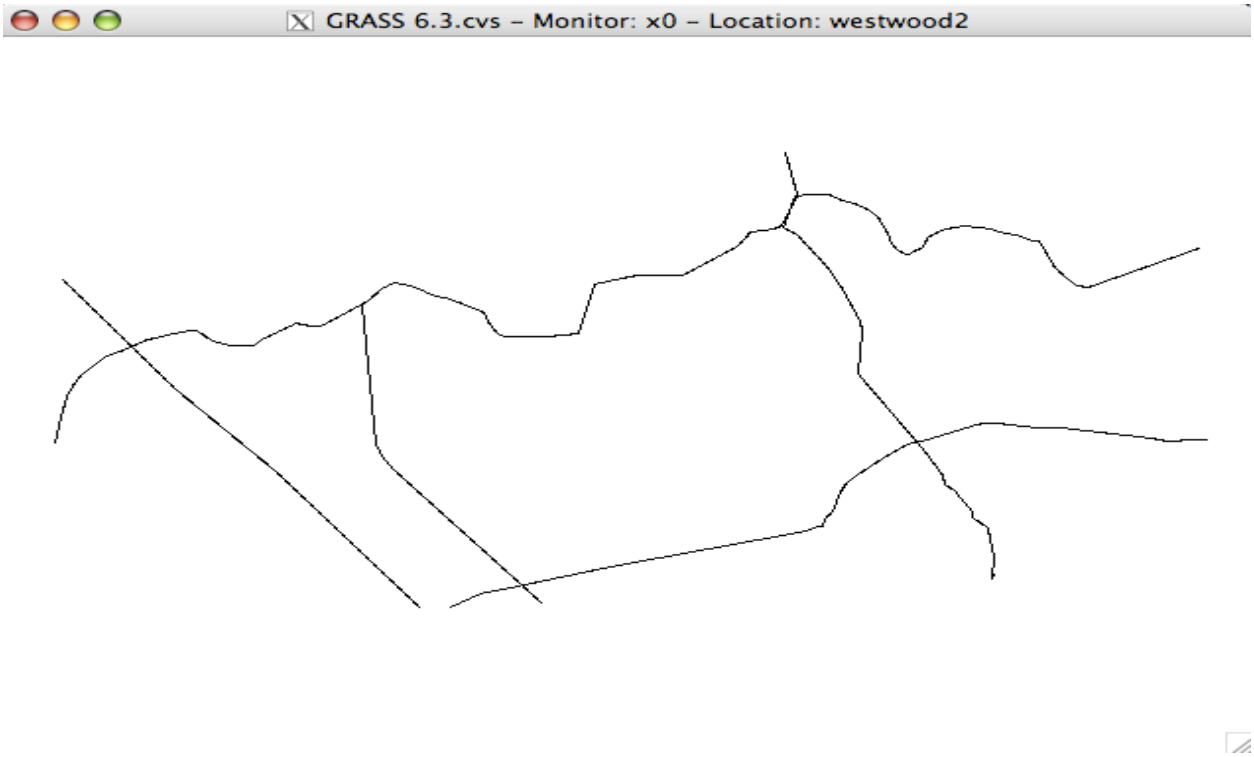
```
http://grass.ibiblio.org/gdp/html_grass63/
http://grass.ibiblio.org/gdp/html_grass64/
```

Examples of raster and vector data:

Elevation map:

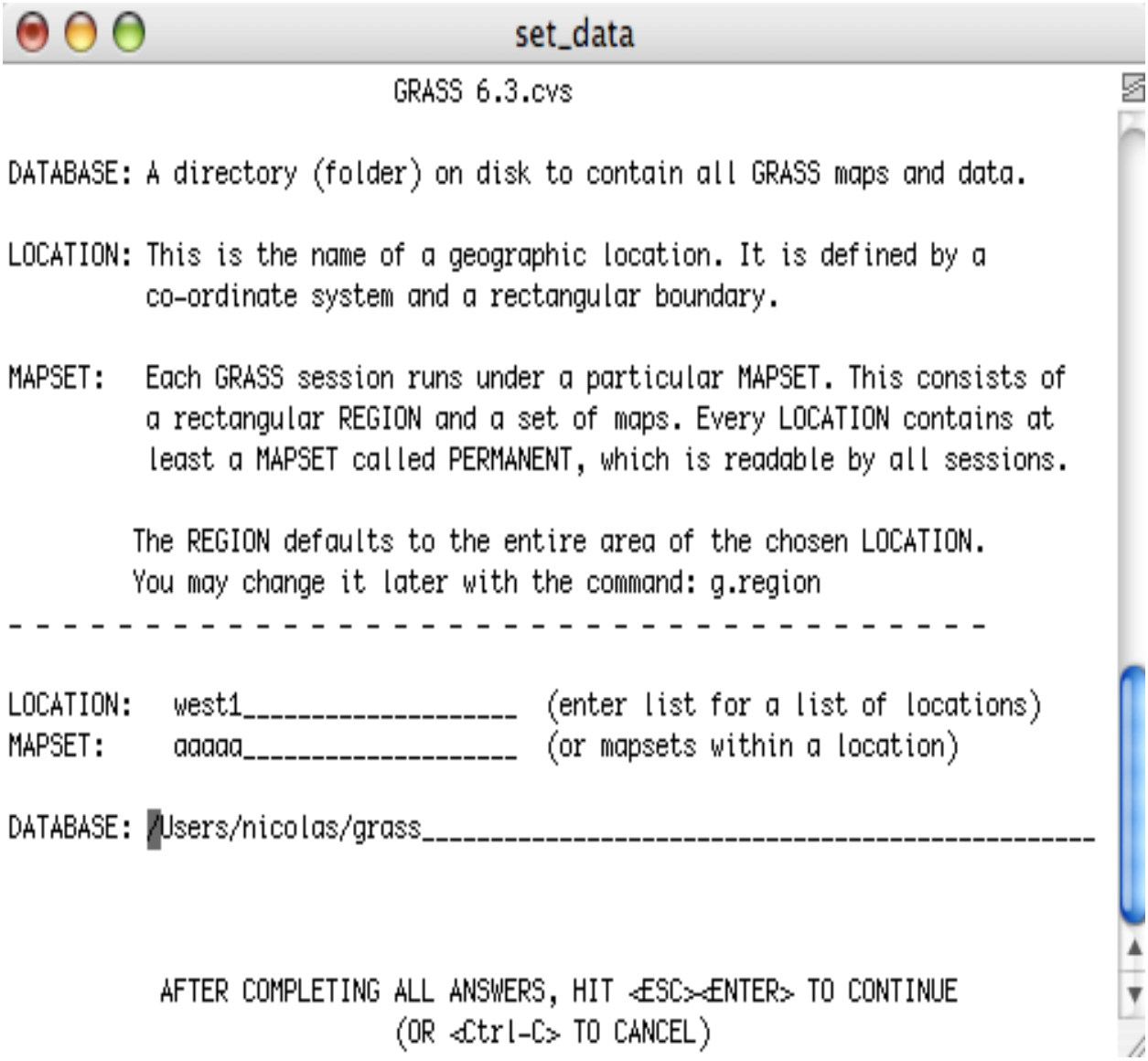


A vector map:



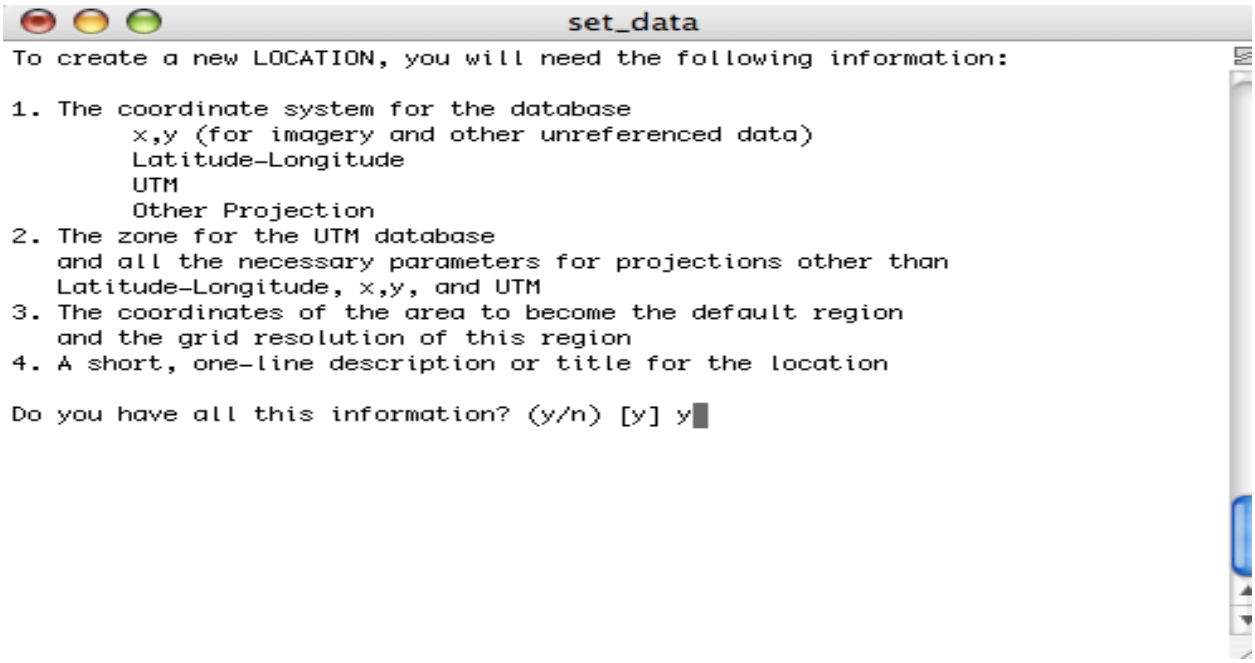
- Before we begin using the GRASS commnands we will first learn how to create a new LOCATION. There are three ways to create a new location (by clicking on **Georeferenced file**, or **EPSG codes**, or **Projection values** as shown on the screenshot below. We will begin using the last one (Projection values).

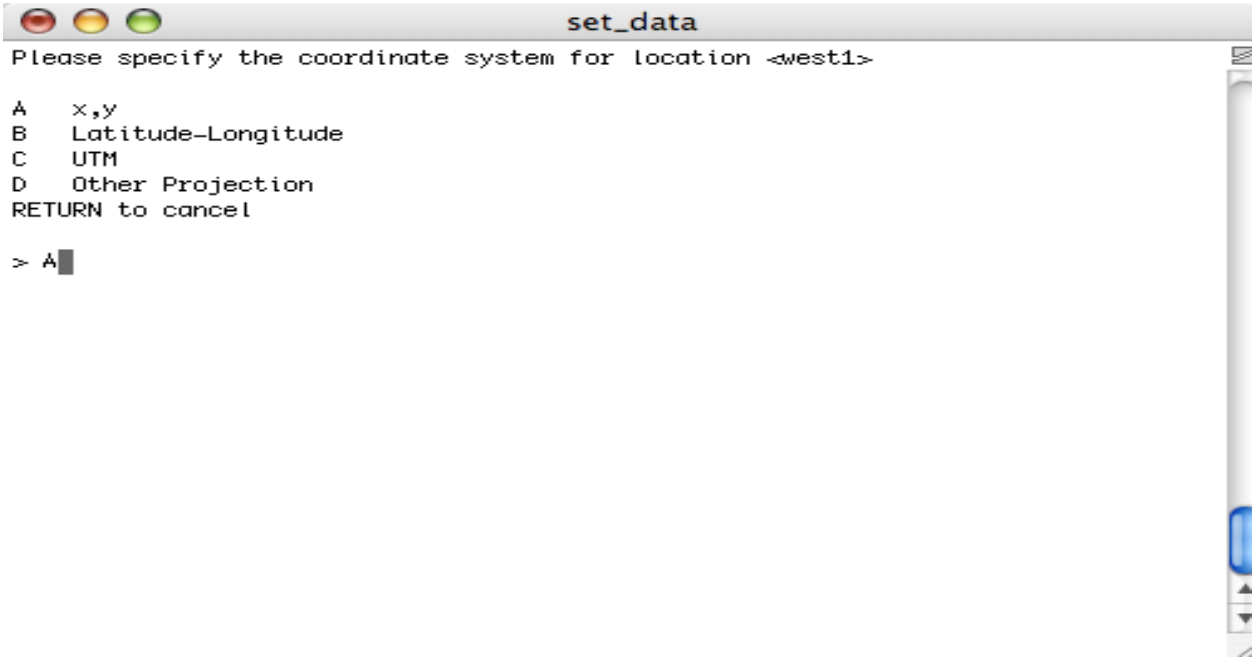
To create a new LOCATION using the last option, click on "Projection values". This will take you to the next window:



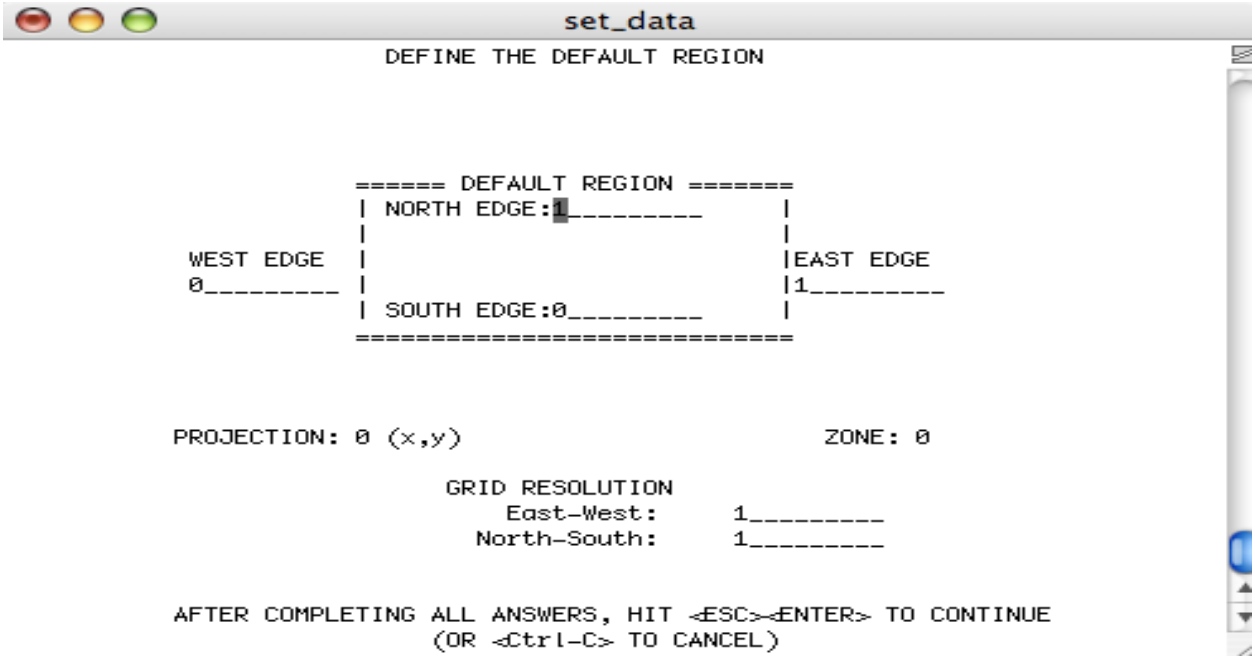
Suppose we want to create the LOCATION "west1". Simply enter the name `west1` next to LOCATION (as shown above) and hit return. Give a name to mapset (say, `aaaaa`). Then press ESC and then ENTER and follow the steps below as described by the screenshots...


set_data
LOCATION <west1> - doesn't exist
Available locations:
20000
3points.txt
Digitize
Grass Included Tutorials.rtf
HELLO
I5
UCLA
UCLA_temporary
a
asterdem30m.img
e1
spearfish60
ucla1
ucla_map.tiff
westwood
Would you like to create location <west1> ? (y/n) [y] y


set_data
To create a new LOCATION, you will need the following information:
1. The coordinate system for the database
x,y (for imagery and other unreferenced data)
Latitude-Longitude
UTM
Other Projection
2. The zone for the UTM database
and all the necessary parameters for projections other than
Latitude-Longitude, x,y, and UTM
3. The coordinates of the area to become the default region
and the grid resolution of this region
4. A short, one-line description or title for the location
Do you have all this information? (y/n) [y] y

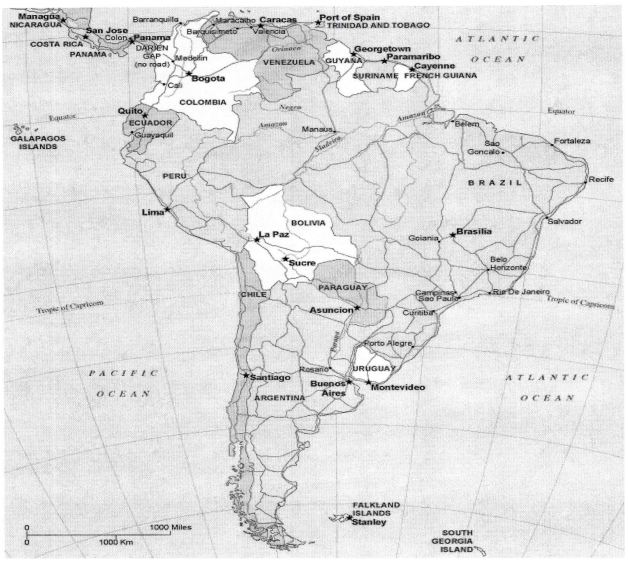


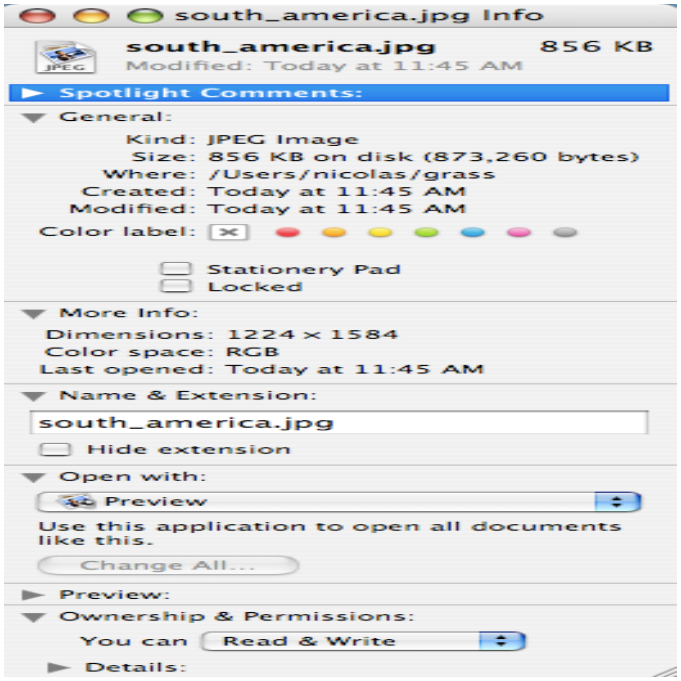
Note: Here we choose the option "A" because we do not have any coordinates available at the moment. We choose a general non-georeferenced coordinate system x, y.



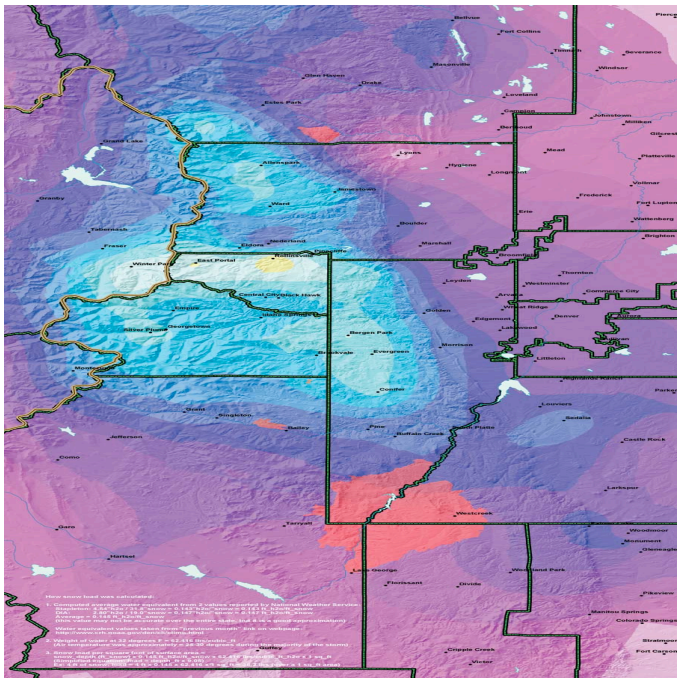
Note: Set the WEST EDGE and SOUTH EDGE values to zero, and enter for NORTH EDGE the number of rows and for EAST EDGE the number of columns. The next page explains what the number of columns and rows are. For the GRID RESOLUTION you can choose 1, because the units are pixels. See next page...

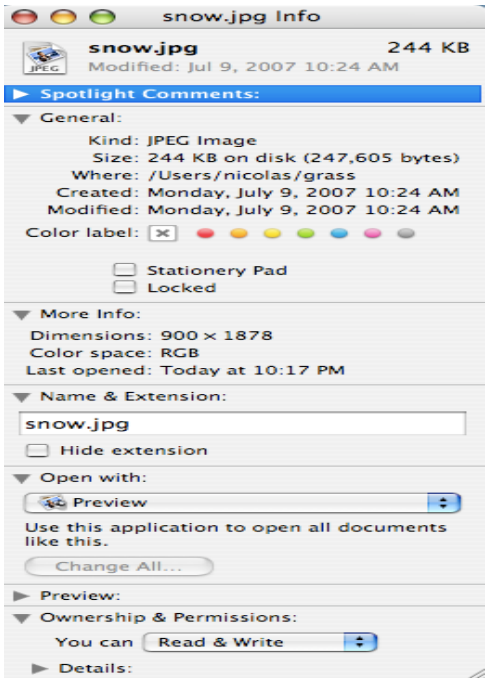
Note: To find the number of rows and columns: If you are using a MAC computer, click on *Apple key+i*. If you are using a PC computer, do right click and select properties. The example below shows the map of South America and the corresponding number of rows and columns. The number of rows and columns that we enter (see previous snapshots) should cover at least the size of the map that we want to import, It can be defined larger than needed. This will not use extra memory since the memory used depends only on the actual size of the file imported.

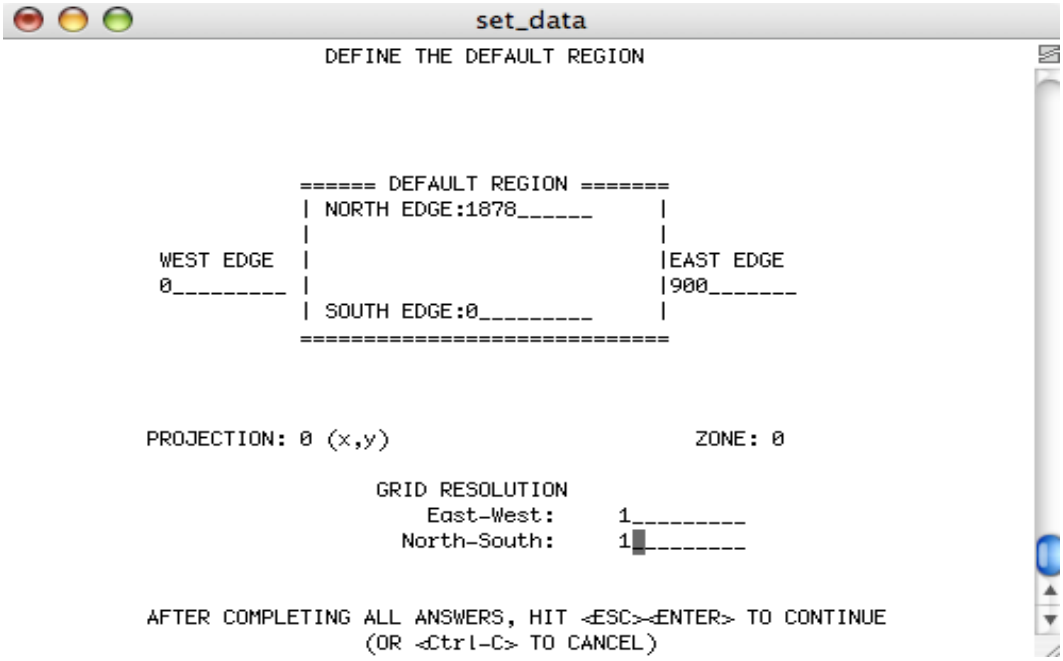


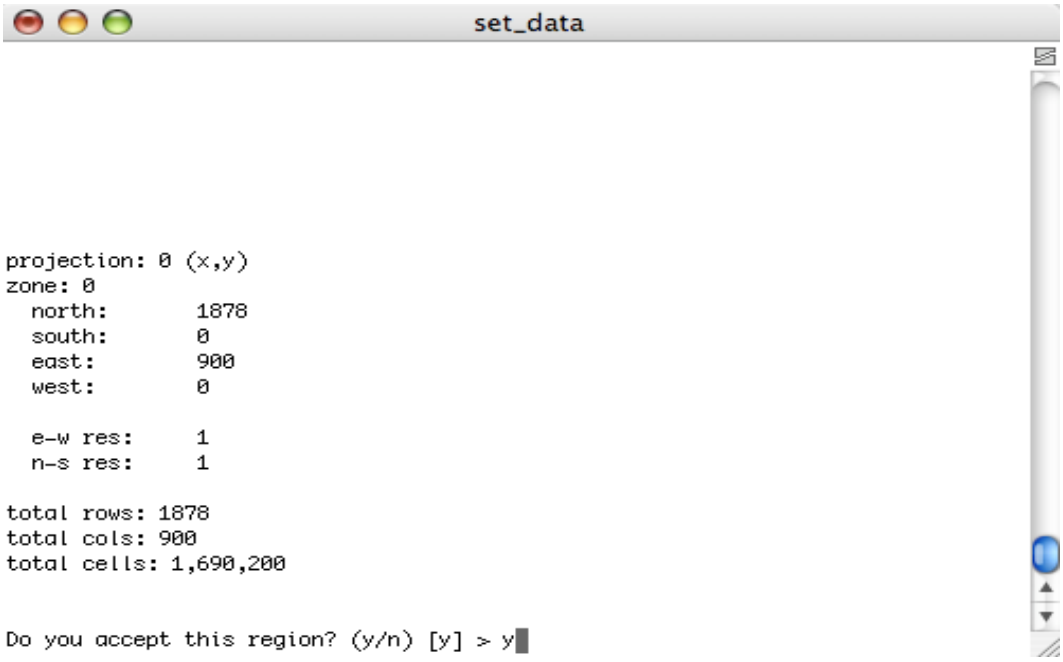


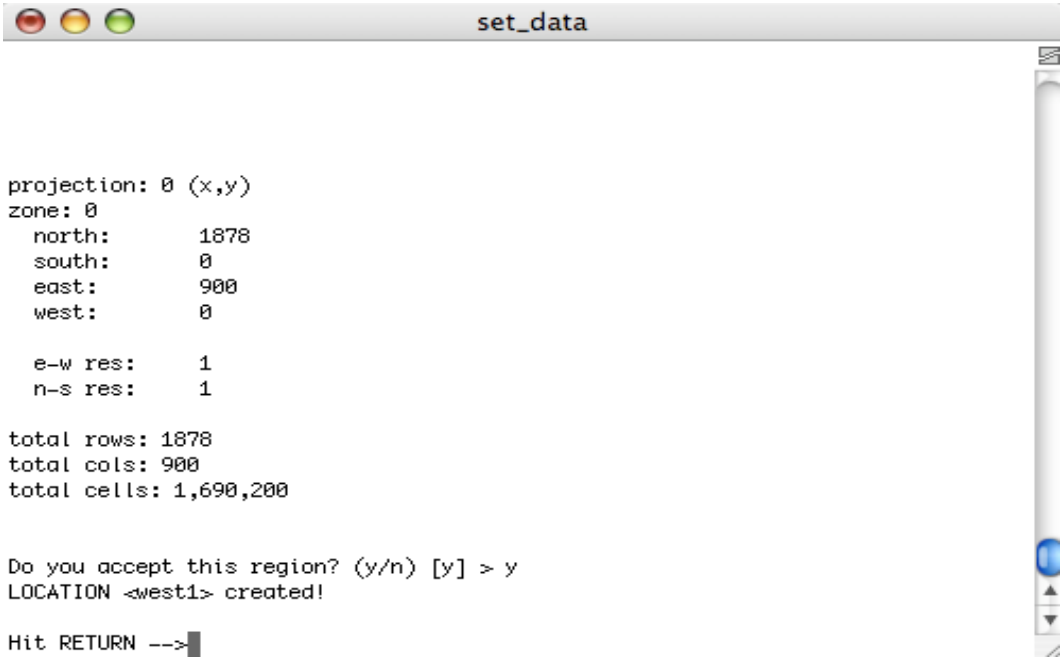
The map we want to import in GRASS for this tutorial is the following:



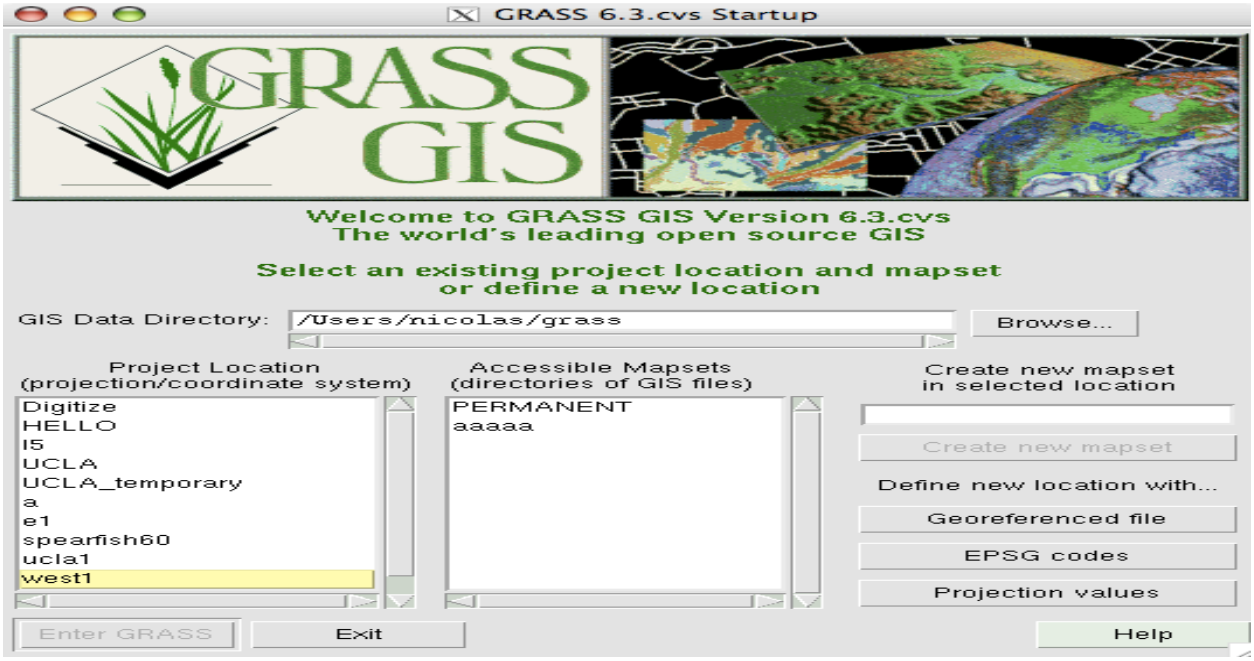



set_data
DEFINE THE DEFAULT REGION
====== DEFAULT REGION =======
| NORTH EDGE:1878_____ |
|
WEST EDGE |
0_____ |
| SOUTH EDGE:0_____ |
============================
EAST EDGE
900_____
PROJECTION: 0 (x,y)
ZONE: 0
GRID RESOLUTION
East-West: 1_____
North-South: 1_____
AFTER COMPLETING ALL ANSWERS, HIT <ESC><ENTER> TO CONTINUE
(OR <Ctrl-C> TO CANCEL)

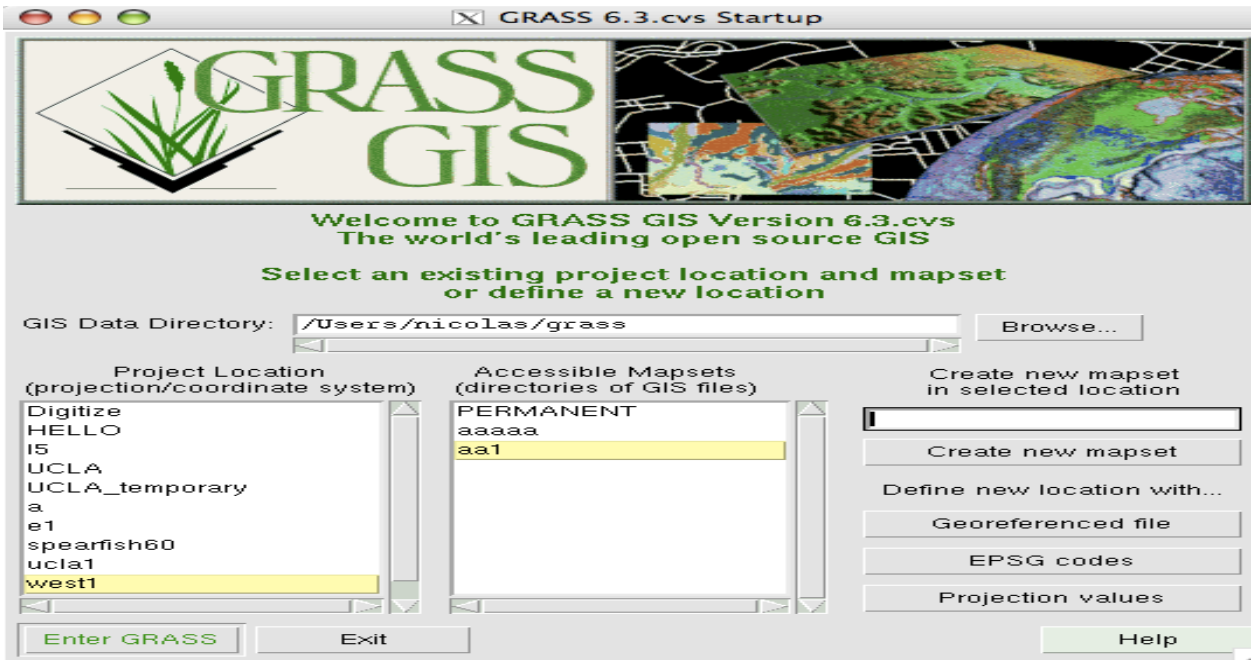

set_data
projection: 0 (x,y)
zone: 0
north: 1878
south: 0
east: 900
west: 0
e-w res: 1
n-s res: 1
total rows: 1878
total cols: 900
total cells: 1,690,200
Do you accept this region? (y/n) [y] > y



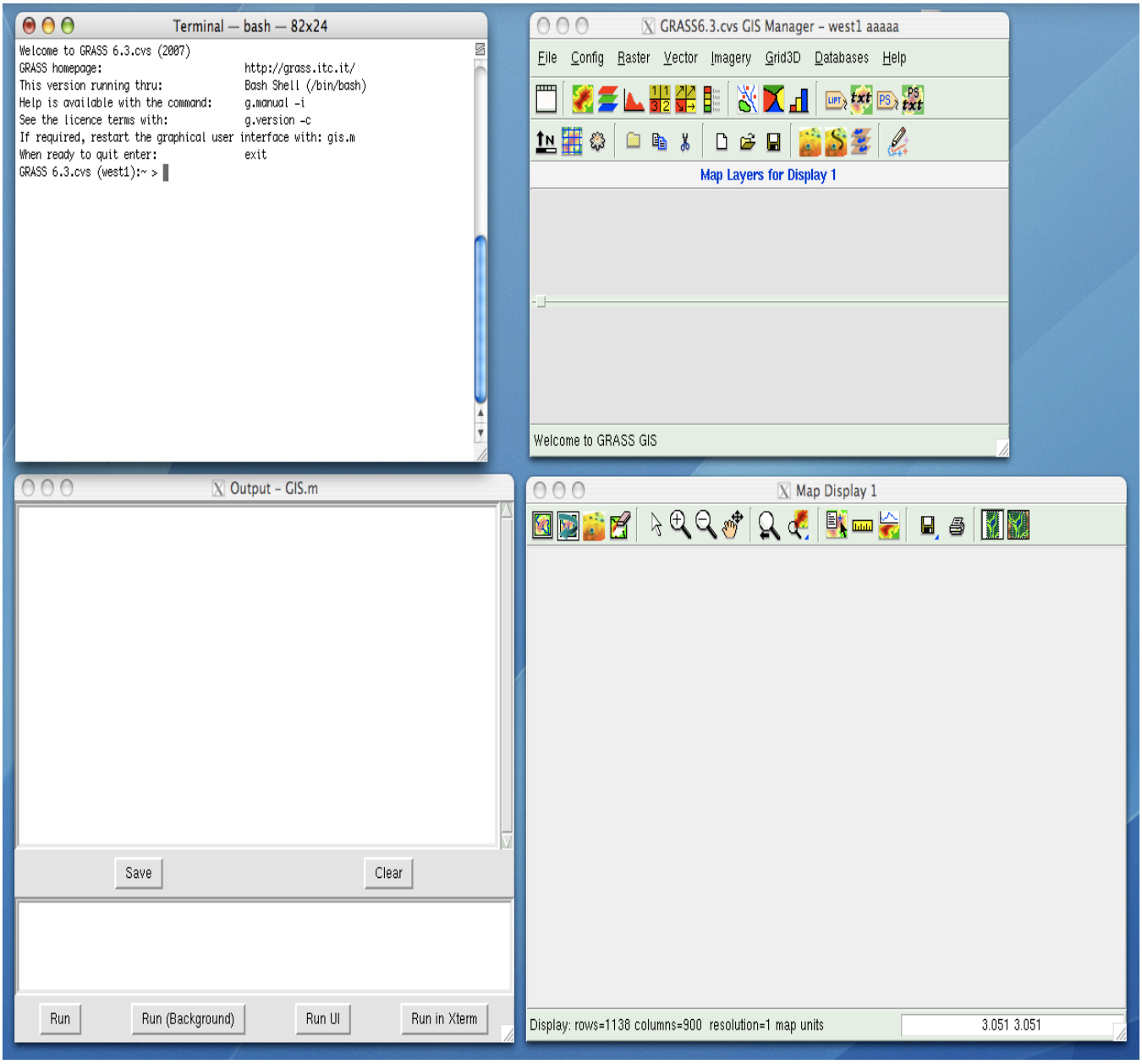
So far, we have created a new LOCATION (west1) and a new MAPSET (aaaaa). Also, GRASS has created automatically the PERMANENT mapset. See next snapshot of the GRASS startup screen.



We can now create new MAPSETs by first selecting the LOCATION we want (here, west1) and then entering the MAPSET's name in the box below "Create new mapset in selected location". Say, we want to create a new mapset named aa1. Here is the output:



To enter GRASS, we select the mapset "aaaaa" and click on "Enter GRASS". This will take you to the following:



Note: The upper left window as shown in this snapshot, is the terminal "GRASS Command Window". This is the window where you type your commands (more will follow soon...). The upper right window it is called the "GIS Manager" and the window on the lower right is the "Map Display" window. The window on the lower left contains messages after we run a GRASS command. The last three windows are linked together. The "GRASS Command Window" can also display maps, etc., but we will have to open a monitor first. So, there are two ways to work with GRASS data: Either through the GRASS Command Window and a monitor, or with the other three windows as shown in the snapshot above.

Import and display raster data

Suppose we want to import and display the map "snow.jpg", which located under

`~/grass/snow.jpg`

On the terminal GRASS Window Command type the following commands (in the snapshots below, we can see the output of these commands):

```
GRASS 6.3.cvs (w1):~ > r.in.gdal in=~/grass/snow.jpg out=snow
Projection of input dataset and current location appear to match.
Proceeding with import...
 100%
 100%
 100%
GRASS 6.3.cvs (w1):~ > g.list rast
```

```
----------------------------------------------
<raster> files available in mapset <a3>:
snow.blue  snow.green snow.red


----------------------------------------------
GRASS 6.3.cvs (w1):~ > d.mon x0
using default visual which is TrueColor
ncolors: 16777216
Graphics driver [x0] started
GRASS 6.3.cvs (w1):~ > d.rast snow.green
 100%
GRASS 6.3.cvs (w1):~ > r.composite
GRASS 6.3.cvs (w1):~ > d.rast snow3
 100%
```

Here is what each one of the previous GRASS commands does:

`> r.in.gdal in=~/grass/snow.jpg out=snow`
Imports the map snow.jpg and gives it the name "snow".

`> g.list rast`
Lists all the maps created by GRASS. Here we have snow.blue, snow.green, snow.red.

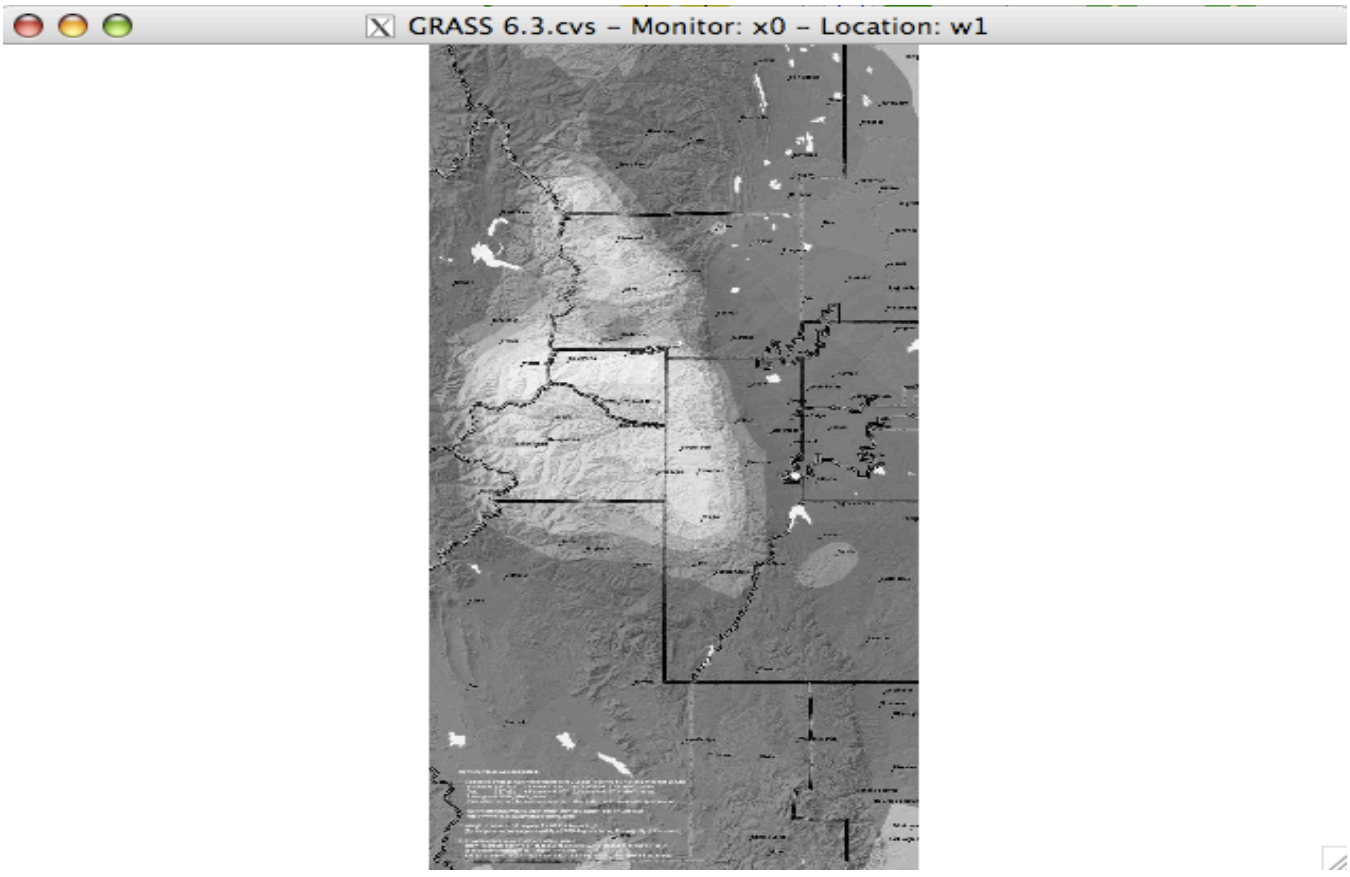
`> d.mon x0`
Opens a new window in order to display the map.

`> d.rast snow.green`
Displays the map snow.green (see screenshot below).

`> r.composite`
Opens a new window in order to combine the three maps (see screenshot below).

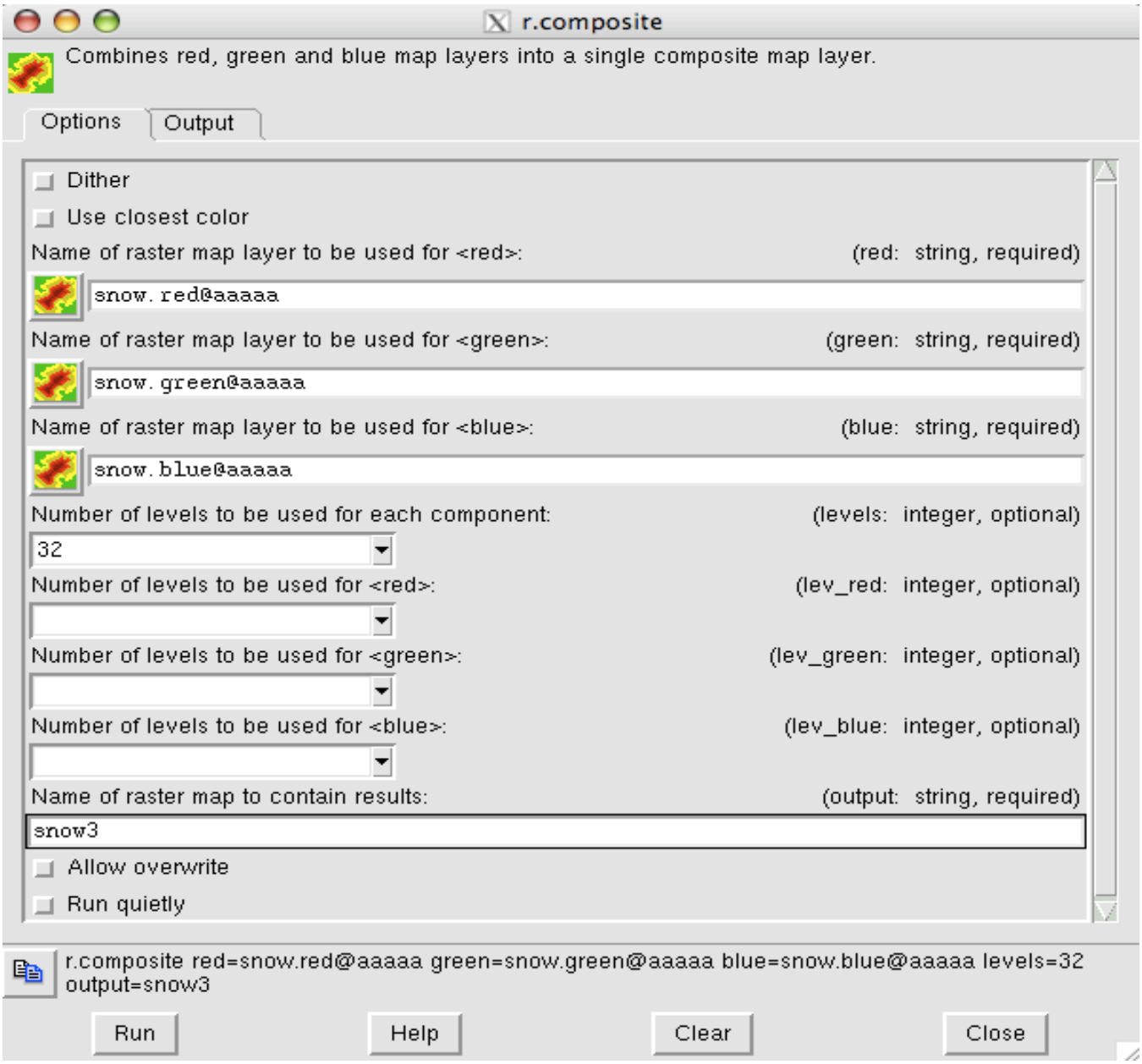
`> d.rast snow3`
Displays the map with the three colors combined.

Map snow.green is displayed:

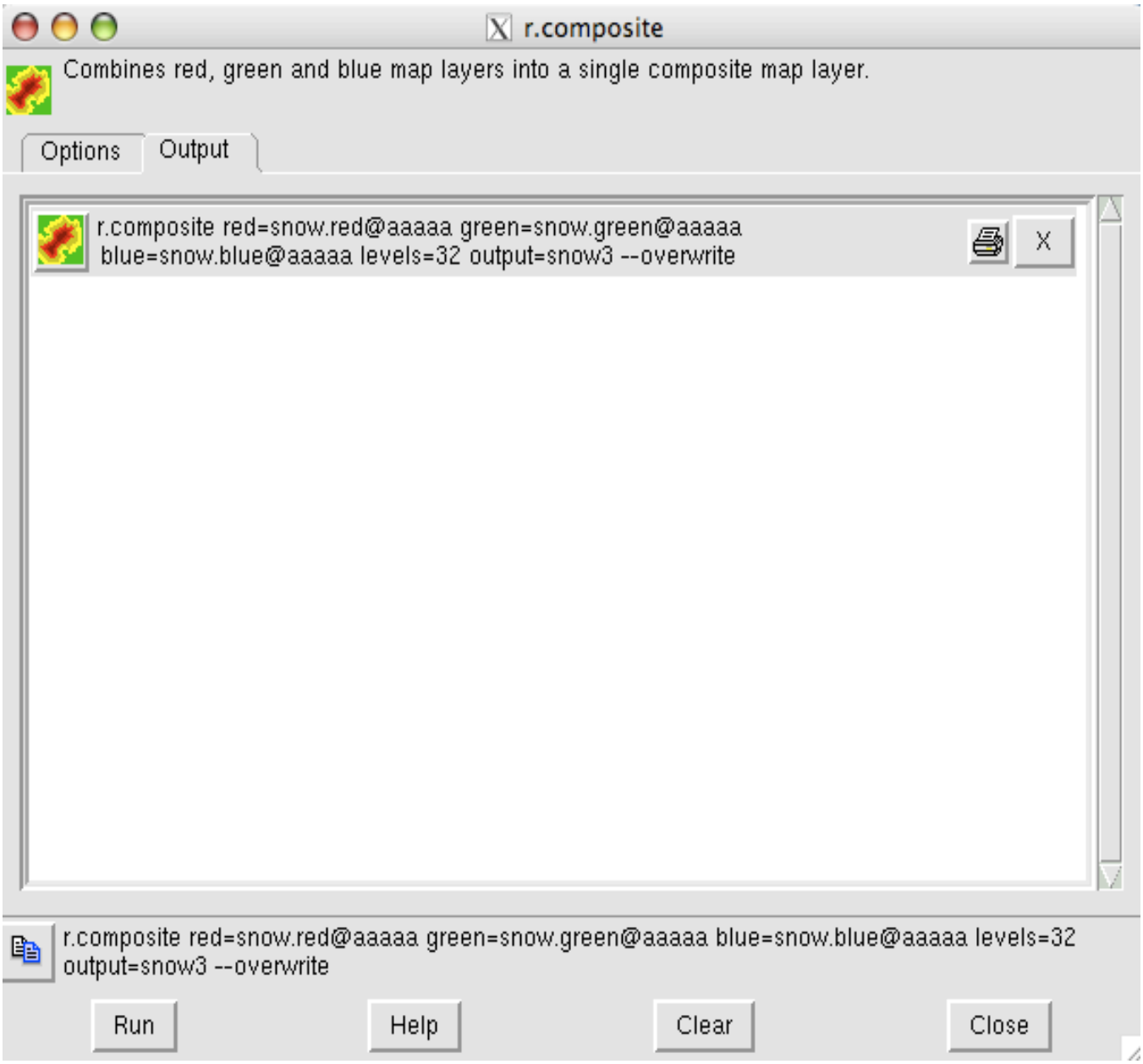


Note: GRASS uses the RGB color model. In this model the three colors (red, green, blue) are used in varying intensities to produce other colors. The intensity of each of the three colors (red, blue, green) is measured on a scale from 0-255 (0 represents no color, 255 represents maximum intensity). For example, the black color can be obtained when $R = 0, B = 0, G = 0$, etc.

Combine snow.blue, snow.green, snow.red into a map named snow3:

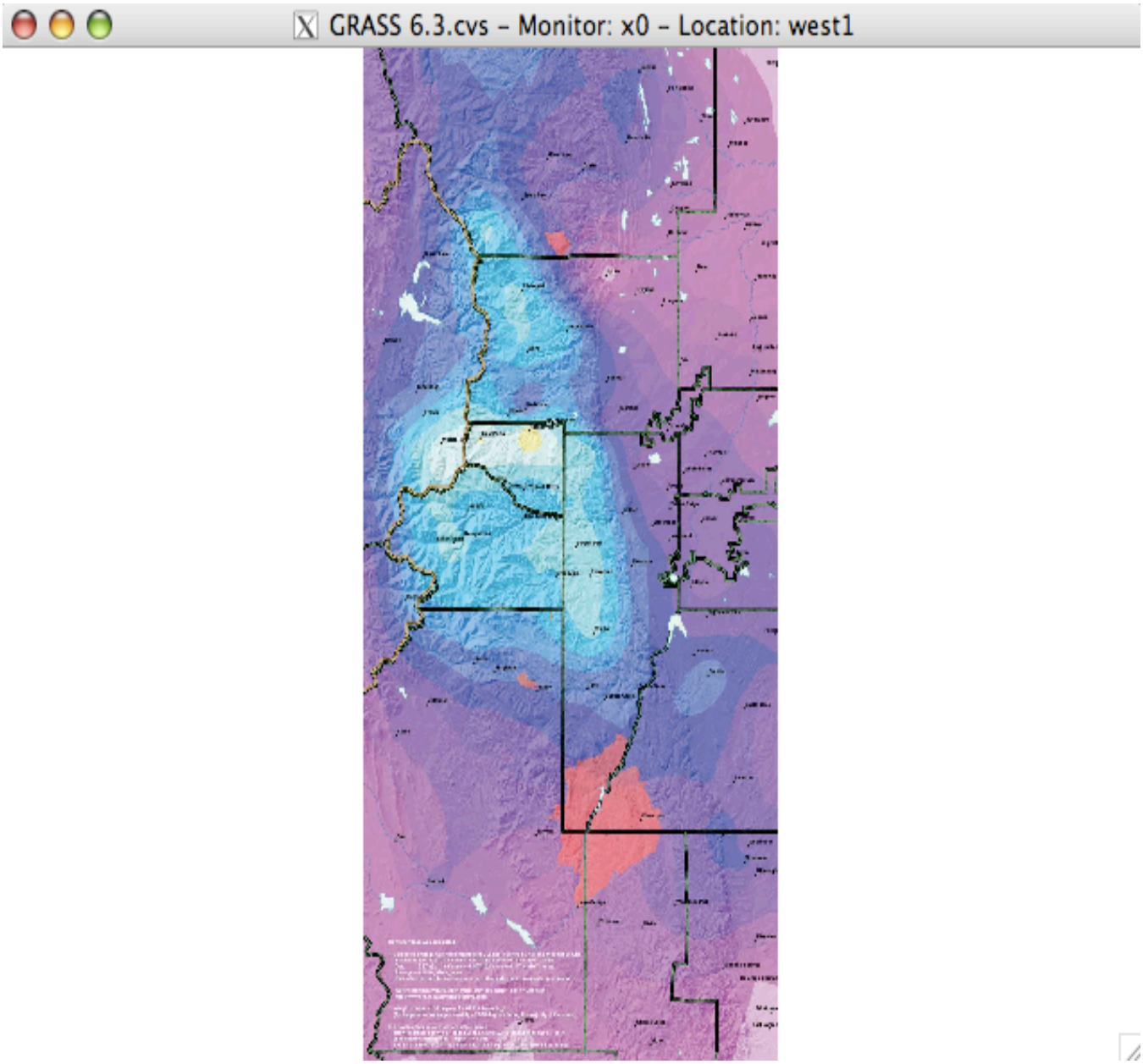


The output:

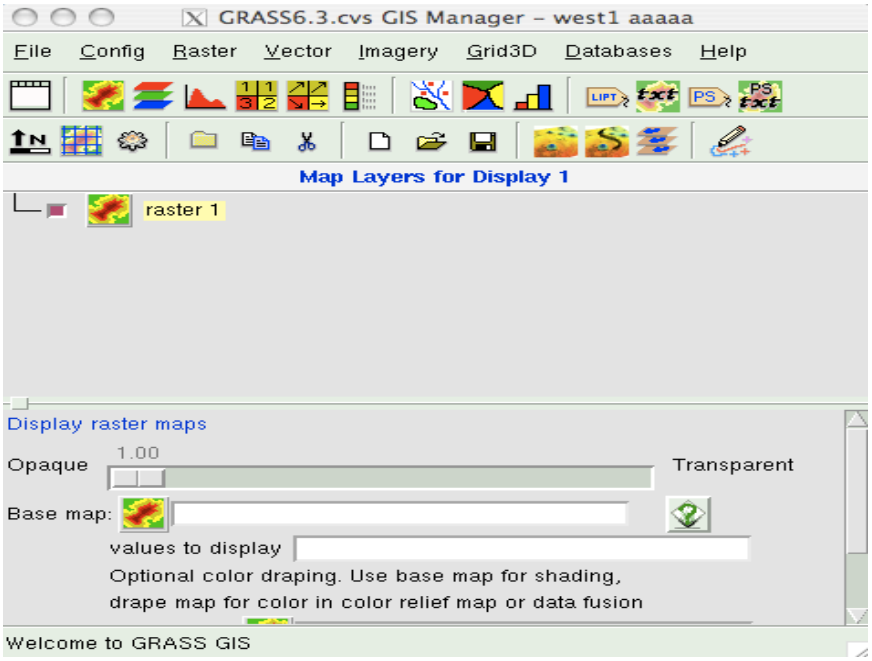

r.composite
Combines red, green and blue map layers into a single composite map layer.
Options
Output
r.composite red=snow.red@aaaaa green=snow.green@aaaaa
blue=snow.blue@aaaaa levels=32 output=snow3 --overwrite
X
r.composite red=snow.red@aaaaa green=snow.green@aaaaa blue=snow.blue@aaaaa levels=32
output=snow3 --overwrite
Run
Help
Clear
Close

The new map:

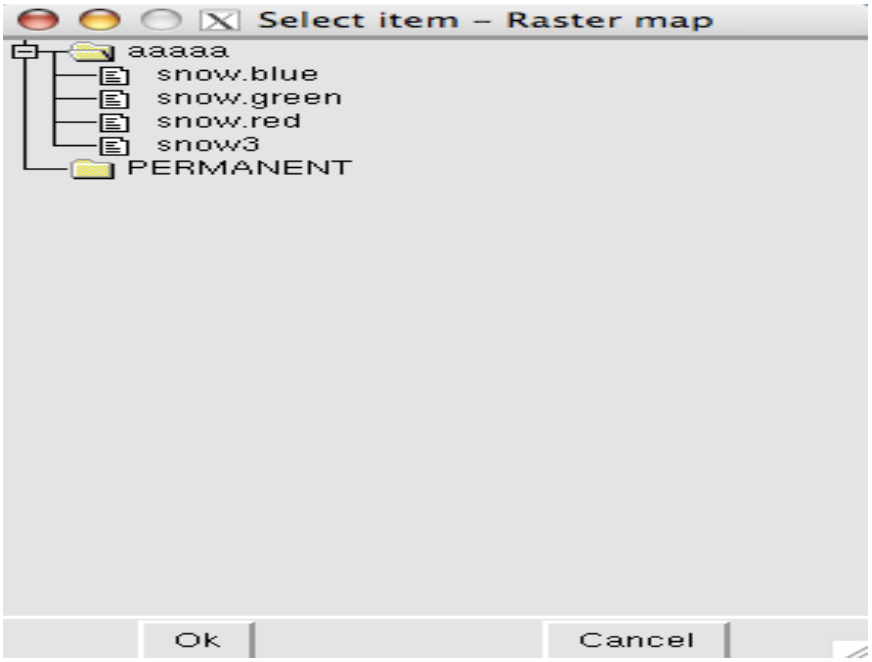
```
> d.rast snow3
```



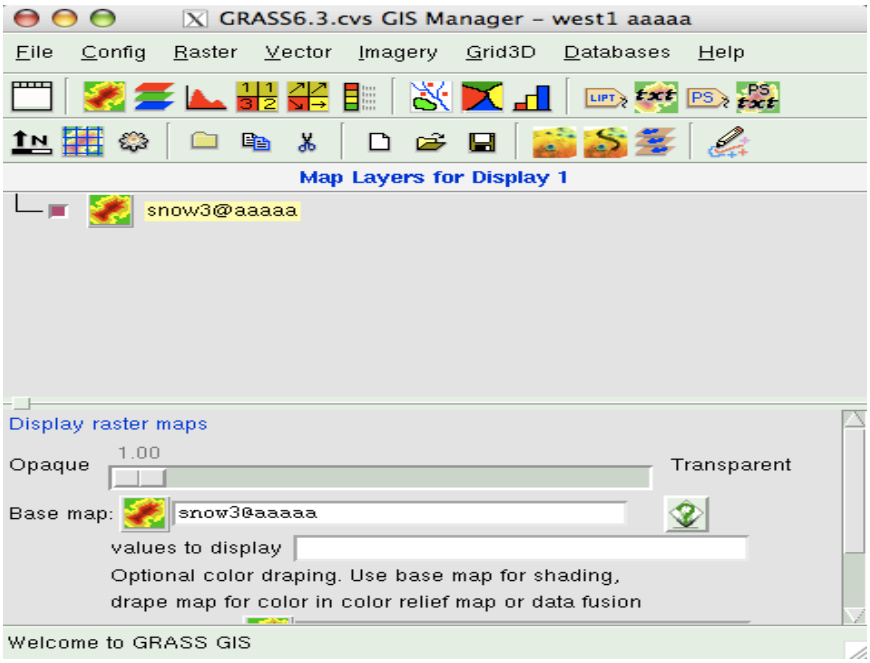
Another way to display maps is through the GIS Manager and the Map Display window. This is shown below in the 4 snapshots. First click on the raster icon below "Config".



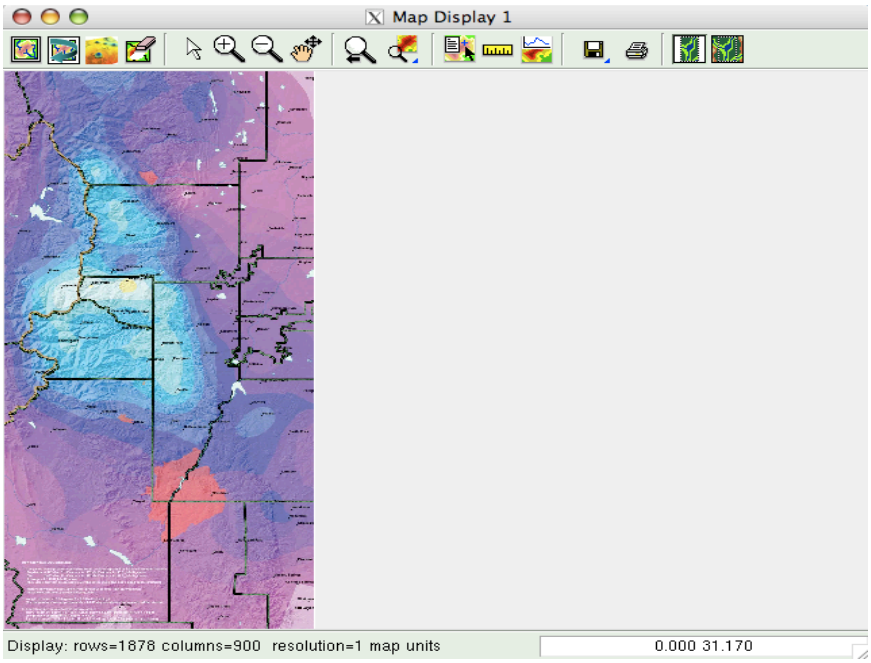
Then click on "Base map:" below and select the raster map snow3.



And looks like this:



And you can display the map at the Map Display window (click on the upper left icon):



The g.region command: Managing map regions and resolutions

- To change the region, resolution, and boundaries of the map the *g.region* command is used with the following flags and options: *-d, -p, -l, -e, -c, -o, -dp, res, n, s, w, e, save.*

-p	Print region
-d	Gives Default region
-l	Gives Print lat/long
-e	Gives extent of the region
-c	Gives center coordinates
save	Saves the current region
res=	Changes resolution
nsres=	Changes resolution n-s
ewres=	Changes resolution e-w
n=	Changes the north extent
s=	Changes the south extent
e=	Changes the east extent
w=	Changes the west extent

Here are some examples:

- Here is how we can get the current region numbers:

```
GRASS 6.3.cvs (west1):~ > g.region -pec rast=snow3
projection:         0 (x,y)
zone:               0
north:              1878
south:              0
west:               0
east:               900
nsres:              1
ewres:              1
rows:               1878
cols:               900
cells:              1690200
north-south extent: 1878.000000
east-west extent:   900.000000
center easting:     450.000000
center northing:    939.000000
GRASS 6.3.cvs (west1):~ >
```

- Suppose we want to select a particular region:

```
GRASS 6.3.cvs (west1):~ > g.region n=1000 e=500 rast=snow3  -pec
projection:         0 (x,y)
zone:               0
north:              1000
south:              0
west:               0
east:               500
nsres:              1
ewres:              1
rows:               1000
cols:               500
cells:              500000
north-south extent: 1000.000000
east-west extent:   500.000000
center easting:     250.000000
center northing:    500.000000
GRASS 6.3.cvs (west1):~ >
```

- We can go back to the default region as follows: `g.region rast=snow3` or

```
GRASS 6.3.cvs (west1):~ > g.region -dp
projection: 0 (x,y)
zone:       0
north:      1878
south:      0
west:       0
east:       900
nsres:      1
ewres:      1
rows:       1878
cols:       900
cells:      1690200
GRASS 6.3.cvs (west1):~ >
```

- We can change the resolution of the region:

```
GRASS 6.3.cvs (west1):~ > g.region nsres=2 ewres=2 rast=snow3 -pec
projection:         0 (x,y)
zone:               0
north:              1878
south:              0
west:               0
east:               900
nsres:              2
ewres:              2
rows:               939
cols:               450
cells:              422550
north-south extent: 1878.000000
east-west extent:   900.000000
center easting:     450.000000
center northing:    939.000000
GRASS 6.3.cvs (west1):~ >
```

- There are many options to set the region. Here is another one: The north and east extent are given in terms of the south and west. The result will be a region on the south-west corner of the current region.

```
GRASS 6.3.cvs (west1):~ > g.region n=s+500 e=w+500 rast=snow3 -p
projection:         0 (x,y)
zone:               0
north:              500
south:              0
west:               500
east:               900
nsres:              1
ewres:              1
rows:               500
cols:               400
cells:              200000
north-south extent: 500.000000
east-west extent:   400.000000
center easting:     700.000000
center northing:    250.000000
GRASS 6.3.cvs (west1):~ >
```

- The changed region can be saved using *g.region -s*, but this must be done from the PERMANENT MAPSET.