Stichting Onderzoek Wereldvoedselvoorziening van de Vrije Universiteit

Centre for World Food Studies

User manual for the SAS-facility to plot maps

by

Bart van den Boom and Saket Pande

Staff Working Paper **WP - 07 - 03** **November 2007**

Contents

1. Introduction

In recent years spatial analysis has gained considerable attention at SOW-VU and the plotting of maps is increasingly used to present research outcomes. This note discusses the facility that has been developed for this plotting using the SAS software (SAS Institute Inc., 2003). It is linked to the gridding, regression, classification and polling software (Keyzer, 2005) but can also be used as a standalone package. It is an extension and revision of an earlier version of the facility (Overbosch, 2006). The main revision concerns the construction of (administrative) boundaries to overlay the basic raster map of grid cells. Whereas these boundaries had to be provided as a separate data set, they are now identified directly from the basic raster map itself following the edges of grid cells at the boundary of a specified (arbitrary) grouping. The extension of the facility consists of additional color schemes and added options to position the legend, to scale the legend and the plot, and to plot at higher or lower resolutions. Also, the parts of the facility that used to be specific to a certain application have been eliminated whereby the facility is now generic and can be initialized and tested through a simple batch-file.

The facility employs a grid with a pre-specified resolution and the grid cells that lie in the area of interest (say, a particular country) are aggregated at three (administrative) levels. The lowest level is indicated by a variable named `CN` (say, county number), the middle level by `PV` (say, province number), and the highest level by `R` (say, region number). Next to this aggregation from low to high administrative levels, grid cells are aggregated according to some other classification, indicated by a variable named `ML` (say, a number that distincts mainland from sea routes, or sandy soils from clay, or urban from rural area). Like the variables `CN`, `PV` and `R`, the variable `ML`-code is an integer-valued code that can take one, two or many values, depending on the application.

In Section 2 we explain how to construct and configure the data files necessary for using the plotting facility. In Section 3 we describe and illustrate the actual use, showing plots that use the default configuration of the options. Section 4 illustrates the option to plot two variables on a single map[1]. Finally, Section 5 briefly describes the various other options of the facility in order to dovetail the plot to specificities. In particular, the facility has options to change the formatting and the positioning of the legends, to choose from a range of color schemes, to add annotates, to scale the plot in order to have more space for legends and titles, to plot at higher resolutions, and, in the case of two variables, to specify their own formatting, their own legend and their own color scheme.

[1] When putting two variables in one plot they should each cover different grid cells (in case of overlap, one of the two must cannot be plotted). Data values for both variables may be stored in a single vector with two dichotomous ranges. In that case overlap is excluded by construction.

2. Construction and configuration of data files

2.1 Preliminary

The facility operates from a folder named `MakeMap` which must have the following structure[2]:

```
MakeMap\SASdat\     : folder containing map data in SAS-format (e.g. locat.sas7bdat)
MakeMap\GRCPdat\    : idem, in GRCP-format (e.g. locat.grd)
MakeMap\GISdat\     : idem, in GIS-format (locat_CN.asc locat_ML.asc CN_PV_R.csv)

MakeMap\SASjobs\         : folder with SAS-programs for processing maps
MakeMap\SASjobs\macros\  : folder with SAS-macros used for processing and plotting maps

MakeMap\Pict\            : folder with plots produced by the plotting facility
```

The processing and plotting of maps discussed in this user manual are produced by SAS-programs that can be executed consecutively by running the batch file:

```
MakeMap\SASjobs\Make_Map.bat
```

2.2 Georeference, resolution, and size of the map

A map is defined as a collection of grid cells that can be plotted as colored dots, say little square boxes, similar to the pixels of a digital picture, and that may form images after grouping. Therefore, as a starting point and a basis for plotting, we consider a grid/raster map of a variable/attribute `CN` that identifies the grid cells according to their lowest level of grouping, say their administrative aggregation. The map is processed by a Geographical Information System (say, the ILWIS software) that supposedly outputs the `CN`-raster in ASCII-format with a header containing information on the georeference, the resolution, and the size of the map.

The following is an example of a `CN`-raster map.

```
File: 'GISdat\GIS_CN.asc'
     ncols          5
     nrows          3
     xllcenter      -1
     yllcenter      10
     cellsize       0.1
     nodata_value   0
     4 1 2 0 0
     1 1 2 2 0
     0 2 2 2 3
```

[2] The folder is available on the shared disk `E:\bart\GRCP\plot_manual\MakeMap\`. Furthermore, in GRCP-applications the parent folder of `MakeMap` has two other folders where data files are kept: `'..\dat\'` and `'..\wkrun\'`.

With known coordinate and projection system, the georeference of the map can be derived from its lower left corner:

`xllcenter` : longitude of the lower left corner at the pixel center (degrees)
`yllcenter` : latitude of the lower left corner at the pixel center (degrees)

The resolution is derived from the size of its cells:

`cellsize` : length of an pixel edge (degrees)

The size of the raster map is defined by the number of rows and columns:

`ncols` : number of columns
`nrows` : number of rows

Grid cells outside the basic map (i.e. without `CN`-code) are identified by the last line of the header:

`nodata_value` : value to identify grid cells without a value for the attribute

This example defines an area somewhere in northern Ghana with a lower left corner of at 1 degree west of the Greenwich meridian and 10 degrees north of the equator. The longitude and latitude of upper right corner at the pixel center is (-0.6, 10.2) and can be computed by adding the number of columns less one times the cell size to the left corner longitude, and, likewise, the number of rows less one times the cell size to the left corner latitude. Of the 15 grid cells, there are 11 that have a `CN`-code. The remaining 4 cells have an unknown value of `CN` and will be left out of the plotting.

The plotting will be done as if each cell has the same size. The above raster map follows the geographical coordinate system in which the cell size is given in degrees, and hence the actual size (i.e. the distance between two adjoined cells) will vary with the location on the globe. As long as the raster map covers a relatively small area and actual cell sizes show little variation, the plot will look good. However, when this is not the case, the plot of the raster map may start to look strange due to the equal representation of relatively small and relatively large cells, and one may opt for an equal-area projection (Stefanovic, 1996). In the prevailing cases, the plotting facility contains the transformation equations that converts geographical coordinate to the location of a raster map of an equal-area projection with a given centre.

2.3 Grouping of grid cells according to ML-code

The grid cells in the basic map are grouped according to their (administrative) `CN`-code. The software facility also allows for a grouping according to another classification represented by an `ML`-code. The code represents any classification at the level of grid cells and distincts, for example, mainland from sea routes, or, land from the inland waters. Hence the software requires a second raster map that has the same georeference, resolution, and size as the `CN`-map and groups cells in accordance to their `ML`-code. For example:

```
File: 'GISdat\GIS_ML.asc'
     ncols          5
     nrows          3
     xllcenter      -1
     yllcenter      10
     cellsize       0.1
     nodata_value   0
     1 1 1 0 0
     1 2 2 1 0
```

```
0 1 1 1 1
```

2.4 Aggregation the CN-map to PV- and R-level

The `CN`-codes are aggregated to two higher (administrative) levels, the `PV`- and `R`-level. This data must be made available as a `*.csv` file. In the above example the file could read:

```
File: 'GISdat\CN_PV.csv'
     CN,PV,R
     1, 5, 20
     2, 1, 10
     3, 1, 10
     4, 2, 20
```

Also the names of the `CN PV R` -codes should be made available as `*.csv` files. In the above example these files could read:

```
File: 'GISdat\CN_NAME.csv'
     CN,NAME
     1 , County_one
     2 , County_two
     3 , County_three
     4 , County_four

File: 'GISdat\PV_NAME.csv'
     PV,NAME
     1 , Province_one
     2 , Province_two
     5 , Province_five

File: 'GISdat\R_NAME.csv'
     R,NAME
     10 , Region_ten
     20 , Region_twenty
```

Finally, the names of the `ML` codes should also be made available, for example:

```
File: 'GISdat\ML_NAME.csv'
     ML,NAME
     1 , Land
     2 , Lake
```

2.5 Processing the georeferenced data for use by SAS and GRCP-applications

Once the input data described in 2.2, 2.3 and 2.4 above have been put in place, three SAS-programs will produce the files needed to use the plotting facility and link it to FORTRAN and GAMS applications in the GRCP-software.

The first program

```
'SASjobs\Make_Map_Locat.sas'
```

converts the basic grid map from a GIS-raster to a SAS-format[3]:

```
INPUT FILES
'GISdat\GIS_CN.asc'     : map containing CN attribute
'GISdat\GIS_ML.asc'     : map containing ML attribute
'GISdat\CN_PV.csv' : table containing CN PV R aggregation

OUTPUT FILE
'SASdat\locat.sas7bdat' : map containing LAT LON CN PV R ML
'GRCPdat\locat.grd'     : map containing LAT LON CN PV R ML
```

The basic map in SAS-format 'SASdat\locat.sas7bdat' is the pivot of the plotting facility along with its basic annotates delineated below. Note that the map also appears in a third format 'GRCPdat\locat.grd', for more general use in the GRCP-software. In some applications, e.g. the Africa foodaid map, this third format is the starting point, rather than the two *.asc files and the *.csv file. Conversion is then done through a simpler program[4].

Next, a second program

```
'SASjobs\Make_Map_Names.sas'
```

writes the names of the CN PV R ML-codes in SAS format and in GAMS format:

```
INPUT FILES
'GISdat\CN_name.csv'    : table containing CN and NAME
'GISdat\PV_name.csv'    : table containing PV and NAME
'GISdat\R_name.csv'     : table containing R and NAME
'GISdat\ML_name.csv'    : table containing ML and NAME

OUTPUT FILES
'SASdat\CN.sas'         : CN names as SAS format definition
'SASdat\PV.sas'         : PV names as SAS format definition
'SASdat\R.sas'          : R  names as SAS format definition
'SASdat\ML.sas'         : ML names as SAS format definition
'GRCPdat\CN.gms'        : CN names as GAMS set definition
'GRCPdat\PV.gms'        : PV names as GAMS set definition
'GRCPdat\R.gms'         : R  names as GAMS set definition
```

[3] There is also a program 'SASjobs\Make_Map_Raster.sas' that does the reverse, i.e. from converts georeferenced SAS data into a GIS-raster in *.asc format with corresponding aggregation and names for CN PV R and ML . Hence the facility provides a two-way interface between data processing in SAS and in GIS.

[4] Conversion is then done by the program 'SASjobs\Make_Map_Locat_grd.sas' converting the input file 'dat\locat.grd' that has variables LAT LON CN PV R ML and a grid cell identifier to a SAS data file. Note though that for the plotting facility to have a proper georeferencing, the information on (xllcenter yllcenter cellsize) must be made part of the input file.

```
'GRCPdat\ML.gms'        : ML names as GAMS set definition
```

Finally, the more elaborate program

```
'SASjobs\Make_Map_Anno.sas'
```

produces the annotates of the basic map, consisting of the grid-by-grid boundaries of each `CN PV R ML`-code:

```
INPUT FILE
'SASdat\locat.sas7bdat' : map containing LAT LON CN PV R ML

OUTPUT FILES
'SASdat\anno_CN.sas7bdat'    : map delineating area with same CN-code
'SASdat\anno_PV.sas7bdat'    : map delineating area with same PV-code
'SASdat\anno_R.sas7bdat'     : map delineating area with same R-code
'SASdat\anno_ML.sas7bdat'    : map delineating area with same ML-code
```

The program uses a dedicated algorithm that identifies cell-by-cell whether the cell is at the outer boundary of the respective `CN PV R ML` and then draws the line segments accordingly.

Once the basic map and its `CN PV R ML` names and annotate files have been constructed, the plotting facility can be invoked to plot the grid cell values of a variable. For its proper functioning, the facility requires data records to be sorted first by latitude and next by longitude.

3. Plotting raster and administrative maps with options set at their default

3.1 Invocation

Availing of the basic map and its names and annotates, the facility is invoked through a SAS-program that must have the following header:

```
options ls=132 ps=9999 nocenter ;
libname Gdevice0 '..\Makemap\SASdat\' ;
libname mapdat  '..\Makemap\SASdat\' ;
libname library[5] '..\Makemap\SASdat\' ;
%include '..\Makemap\SASdat\CN.sas' ;
%include '..\Makemap\SASdat\PV.sas' ;
%include '..\Makemap\SASdat\R.sas' ;
%include '..\Makemap\SASdat\ML.sas' ;
%include '..\Makemap\SASjobs\macros\map_macros.sas' ;
libname dat  '..\dat\' ;
```

Next, the default plotting of a raster map and an administrative map, respectively, is done through the invocation either of the following macros.

```
%map_grid
  (dataset = [specify SAS-dataset with LAT LON identifier]
  ,variable= [choose a variable from the dataset]
  ) ;
%map_CN_PV_R
  (dataset = [specify SAS-dataset with CN, PV, R or ML identifier]
  ,variable= [choose a variable from the dataset]
  ,CN_PV_R = [specify aggregation level; must be CN, PV, R or ML]
  ) ;
```

The above default invocations will produce a graph with a default name in the folder `'Pict\'`:

```
File: 'Pict\<dataset>_<variable>.gif'
```

where < > indicates the value of the respective argument.

If a format for the plotting variable is available in the library then this format can be used for the coloring of the plot and the drawing of its legend through the optional parameter:

```
  format  = [choose a format from the library].
```

For example, the format of the `CN, PV, R` and `ML` codes is given by their respective names identified in section 2.5, and we may use these formats to plot the administrative map.

[5] In order to make an existing library of SAS-formats available for plotting, the file that contains these formats (e.g. `'formats.sas7bcat'`) must be copied to the folder `'..\Makemap\SASdat\'`.

3.2 Example: Plotting administrative boundaries, possibly with names

By default, plots are annotated with the administrative boundaries at CN, PV and R-level. For example the following produces a plot of CN-values:

```
%map_grid
  (dataset = Mapdat.locat
  ,variable= CN
  ) ;
```

The corresponding plot:

```
File: 'Pict\Mapdat.locat_CN.gif'
```

can be found in Annex 1.

In order to display the names rather than the code, one can use the format= option. For example, the administrative map at the PV-level can be plotted as:

```
%map_grid
  (dataset = Mapdat.locat
  ,variable= PV
  ,format  = PV
  ) ;
```

producing the map:

```
File: 'Pict\Mapdat.locat_PV.gif'
```

To see also the boundaries at the ML-level, one has to use the anno= option to add annotate files. The invocation:

```
%map_grid
  (dataset = Mapdat.locat
  ,variable= ML
  ,format  = ML
  ,anno    = Mapdat.Anno_ML
  ) ;
```

produces the map:

```
File: 'Pict\Mapdat.locat_ML.gif'
```

In the final example of plotting the administrative map we use the scale_ls=0 option which suppresses the default annotates. In this manner, one can plot the R-map showing the own boundaries only:

```
%map_grid
  (dataset = Mapdat.locat
  ,variable= R
  ,format  = R
  ,anno    = Mapdat.Anno_R
  ,scale_ls= 0
  ) ;
```

```
File: 'Pict\Mapdat.locat_R.gif'
```

3.3 Example: Plotting georeferenced data values at grid level

Next consider the plotting of grid-cell-values for a variable. The data are assumed to be available in a SAS dataset in which each record is georeferenced by `LAT LON`. Only records with non-missing data need to be included.

```
File: 'POP'
     LAT LON POP
       1   2 100
       1   3 200
       1   4 250
       1   5 800
       2   1  50
       2   4  50
       3   1   0
       3   2  10
       3   3  20
```

The invocation:

```
%map_grid
  (dataset = POP
  ,variable= POP
  ) ;
```

produces the plot:

```
File: 'Pict\POP_POP.gif'.
```

Note that the SAS dataset can either be created directly by a SAS program or derived from a GIS raster map (`*.acs` file outputted from GIS, as above) or a map outputted from the GRCP-software (`*.csv` or `*.grd` file, sorted by `LAT LON`). For example, the above dataset could have been created either from the file:

```
File: '..\dat\POP.asc'
     ncols         5
     nrows         3
     xllcenter     -1
     yllcenter     10
     cellsize      0.1
     nodata_value  -1
       0  10  20  -1  -1
      50  -1  -1  50  -1
      -1 100 200 250 800
```

or from the file[6]:

```
File: '..\wkrun\POP.csv'
     100
     200
     250
     800
     50
     .
     .
     50
      0
     10
     20
```

3.4 Example: Plotting georeferenced data values at administrative level

Next consider the plotting of administrative data for the variable `POP.` The data are assumed to be available in a dataset in which each record is georeferenced by `CN` and, as before, records with missing data need not be included.

```
File: 'POP_CN'
    CN    POP
     1     60
     2    620
     3    800
     4      0
```

The program that plots these aggregated data proceeds in two steps. The first step merges the data with the basic raster map, using `CN`, `PV`, `R` or `ML` as key. In this example, the key is `CN.` The second step then plots the `CN`-map, as a raster map in which cells belonging to the same district `CN` are given the same data value, and hence the same color. The invocation:

[6] Data to be plotted can also be located in another folder. In particular, while output files of the GRCP-software are given the extension `*.csv` and put into `'..\wkrun\'`, input files generally have the extension `*.grd` and are stored in `'..\dat\'`.

```
%map_CN_PV_R
  (dataset  =pop_CN
  ,variable =POP
  ,CN_PV_R  =CN
  ) ;
```

produces the map:

```
File: 'Pict\POP_CN_POP.gif'
```

4. Plotting raster maps with two variables

4.1 Invocation

The plotting of a raster map with two variables is done through the following invocation.

```
%map_grid
  (dataset  = [specify SAS-dataset with LAT LON identifier]
  ,variable = [choose a first variable from the dataset]
  ,variable2= [choose a second variable from the dataset]
  ) ;
```

Of course, when putting two variables in one plot, they should each cover different grid cells. In case of overlap the value of the first variable will prevail. In some cases, data values of both variables are stored in a single vector with two dichotomous ranges. In that case overlap is excluded by construction and plotting is done through a separate macro:

```
%macro map_grid_combi
  (dataset  = [specify SAS-dataset with LAT LON identifier]
  ,combi_var= [choose the combi-variable from the dataset]
  ,first    = [threshold value between first and second part]
  ) ;
```

As before, in either case, the invocation produces a plot with the default name:

```
File: 'Pict\<dataset>_<variable>.gif'
```

4.2 Example: Plotting two variables on a single map

Suppose the data set from the previous section containing the variable `POP` is expanded with a second variable `NAT` that attains a value only at grid cells where the value of `POP` is missing.

```
File: 'POP_NAT'
     LAT LON POP   NAT
       1   2 100   .
       1   3 200   .
       1   4 250   .
       1   5 800   .
       2   1  50   .
       2   2  .    0.5
       2   3  .    1.8
       2   4  50   .
       3   1   0   .
       3   2  10   .
       3   3  20   .
```

Then the two variables can be plotted on a single map as follows.

```
%map_grid
  (dataset  =POP_NAT
  ,variable =POP
  ,variable2=NAT
  ) ;
```

This produces the plot:

```
File: 'Pict\POP_NAT_POP.gif'
```

4.3 Example: Plotting a variable with two dichotomous ranges

A similar plot can be made when the two variables are stored in a single variable with two dichotomous ranges. Supposing that the threshold value between the first and the second part of the range is set to 10 and that this threshold is to be subtracted from values in the upper range, consider the following data set.

```
File: 'POPNAT'
     LAT LON   POPNAT
       1   2    110
       1   3    210
       1   4    260
       1   5    810
       2   1     60
       2   2     0.5
       2   3     1.8
       2   4     60
       3   1     10
       3   2     20
       3   3     30
```

The invocation is now as follows:

```
%map_grid_combi
  (dataset    = POPNAT
  , combi_var  = POPNAT
  ,first      = 10
  ) ;
```

which produces a plot that resembles the previous one `POP_NAT_POP.gif`:

```
File: 'Pict\POPNAT_POPNAT.gif'.
```

The difference between the two plots is the order in which the two variables appear on the map. The macro `%map_grid_combi` takes the lowest range to be the first variable. This corresponds to the `NAT` variable of the previous example, where it was plotted as the second variable. The use of the corresponding default colors and default position of the legends cause the difference between the two plots `POPNAT_POPNAT.gif` and `POP_NAT_POP.gif`.

The following plot illustrates the options to change the default color schemes and the default position of the legends[7]. In the example at hand, the color and legend of the two parts of the plot can be interchanged as follows.

```
%map_grid_combi
  (dataset  = POPNAT
  ,combi_var= POPNAT
  ,first    = 10
  ,position = 3
  ,color    = BLUE
  ,position2= 1
  ,color2   = GREEN_RED_
  ,graph    = COLOR_POSITION
  ) ;
```

Note that the corresponding plot has been given its own name using the `graph=` option. It can be seen that the map of the combi-variable `POPNAT`

```
File: 'Pict\COLOR_POSITION.gif'
```

is now practically the same as the map `'Pict\POP_NAT_POP.gif'` that plotted the two variables `POP` and `NAT`.

[7] Annex 2 provides the full list of 60 color schemes, while in Annex 3 we discuss the various options for each of the three plotting macros in detail, including the positioning of legends in one of 8 wind directions.

5. Options of the plotting facility

The plotting facility consists of the three SAS-macros discussed and illustrated above. The use of the macro-program that plots data values of one or two variables at the grid cell level:

```
%macro map_grid.
```

has been illustrated in examples in section 3.2, 3.3, and 4.2. The second macro plots data values at `CN`, `PV`, `R`, or `ML`-level:

```
%macro map_CN_PV_R,
```

and has been illustrated in section 3.4. In the case when data values of two variables are stored in a single vector with two dichotomous ranges, the plotting is done by invoking the third and final macro, as illustrated in section 4.3:

```
%macro map_grid_combi.
```

The examples in section 3 and 4 illustrated the use of the facility with minimum required parameters and, in a few cases, the use of optional parameters. For example, in section 3.2, we illustrated the use of the format option to display the `PV`, `R`, or `ML`-names instead of their code, and indicated how to add annotate files and/or how to suppress the default one. In section 4.3 we further illustrated the options to change the color schemes and the position of the legends.

As mentioned in the introduction, the options illustrated in the examples are part of a variety of options. These include the formatting in percentiles, equidistant or user specified groups, the positioning and size of the legends, the titling of the plot (up to 4 lines), the choice of color schemes, the addition and suppression of annotates, the scaling and resolution of the plot, the focusing at the level of selected `CN`, `PV`, `R`, or `ML`-codes, and the distinction between background points and points with missing data. The full list of options for each of the three macros is given and explained in Annex 3.

References

Keyzer M.A. (2005) “Rule based and support vector regression and classification algorithms for joint processing of census, map survey and district data”. SOW-VU Working Paper 05-01. Centre for World Food Studies, Vrije Universiteit Amsterdam.

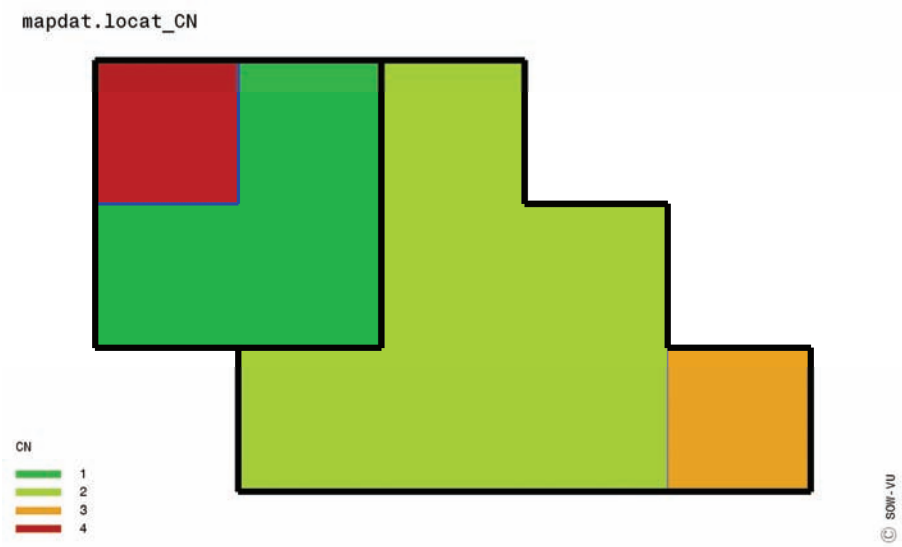
Overbosch G.B. (2006) “'Inside the Map Factory: Note on making maps with SAS'”. SOW-VU Working Paper 06-01. Centre for World Food Studies, Vrije Universiteit Amsterdam.

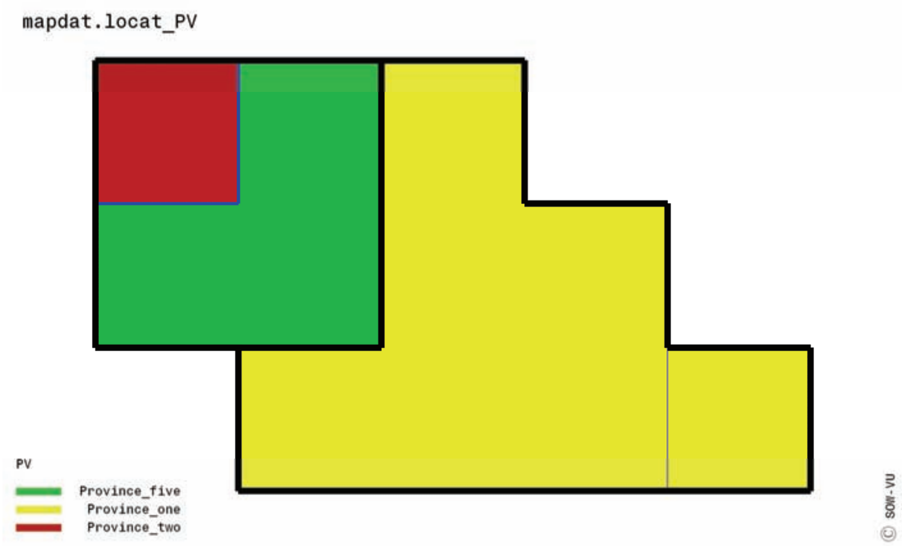
SAS Institute Inc. (2003) SAS OnlineDoc® 9.1. Cary, North Carolina.

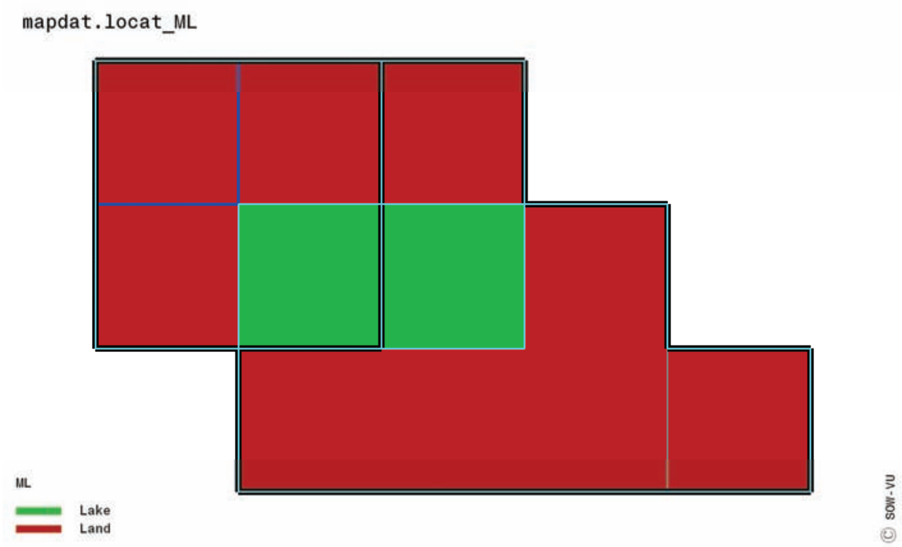
Stefanovic P. (1996) “Georeference and coordinate transformations”. ITC International Institute for Aerospace Survey and Earth Sciences, Enschede.

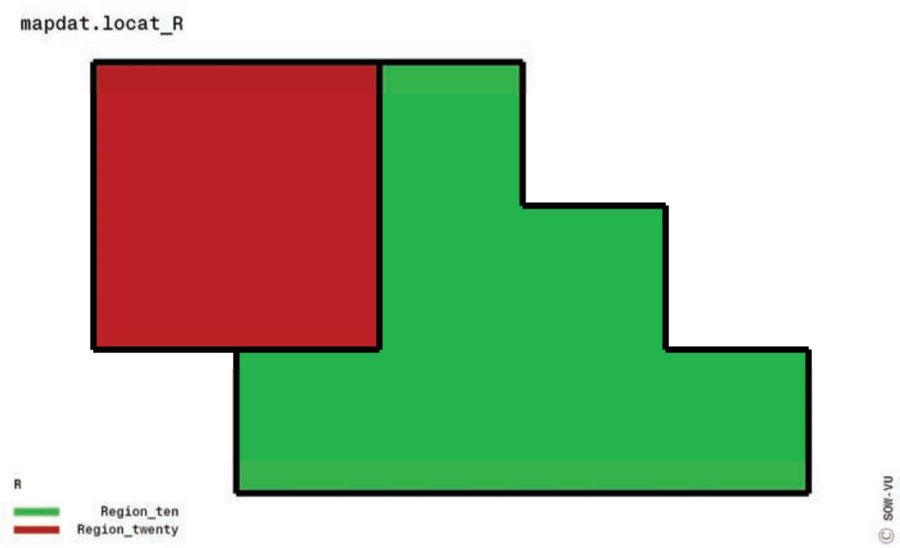
Annex 1. Example plots

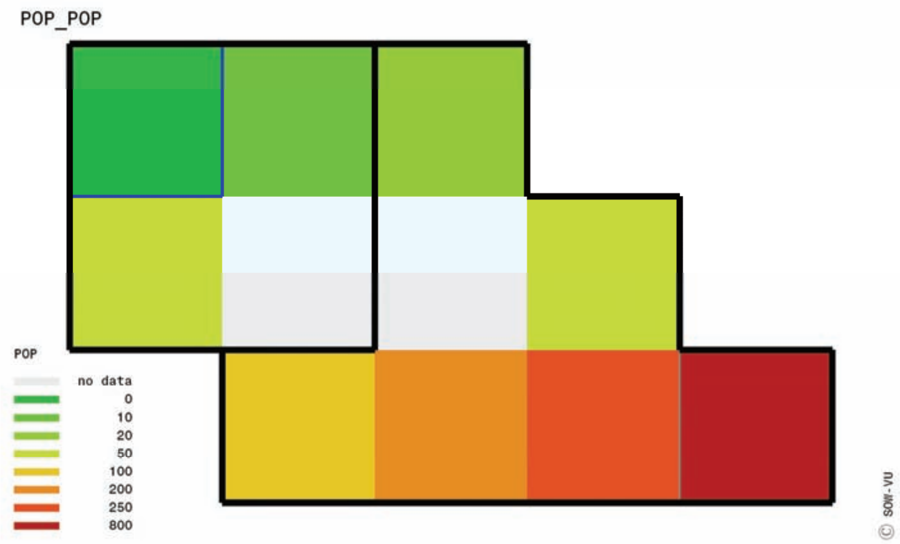
```
From section 3.2:
File: 'Pict\Mapdat.locat_CN.gif'
File: 'Pict\Mapdat.locat_PV.gif'
File: 'Pict\Mapdat.locat_ML.gif'
File: 'Pict\Mapdat.locat_R.gif'

From section 3.3:
File: 'Pict\POP_POP.gif'

From section 3.4:
File: 'Pict\POP_CN.gif'

From section 4.2:
File: 'Pict\POP_NAT_POP.gif'

From section 4.3:
File: 'Pict\POPNAT_POPNAT.gif'
File: 'Pict\COLOR_POSITION.gif'
```

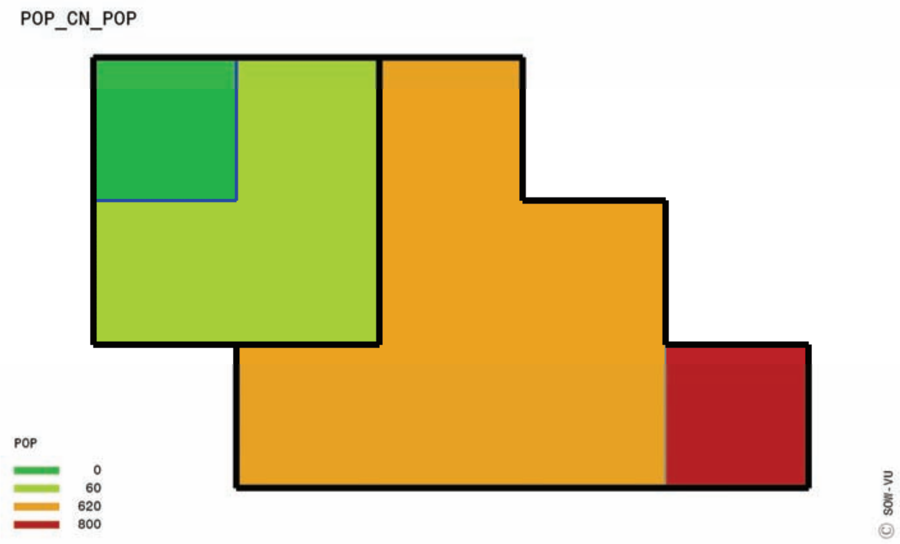

mapdat.locat_CN
CN
1
2
3
4
© SOW-VU

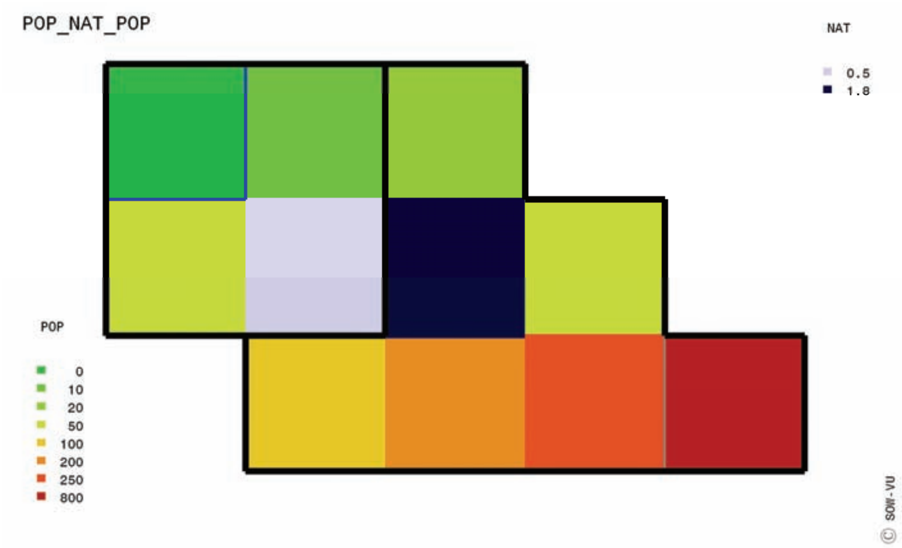

mapdat.locat_PV
PV
Province_five
Province_one
Province_two
© SOW-VU

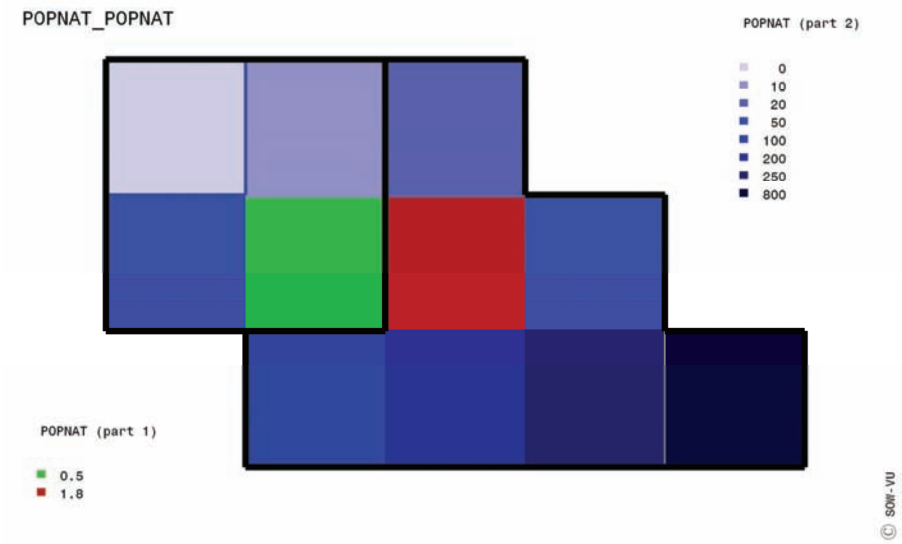

mapdat.locat_ML
ML
Lake
Land
© SOW-VU

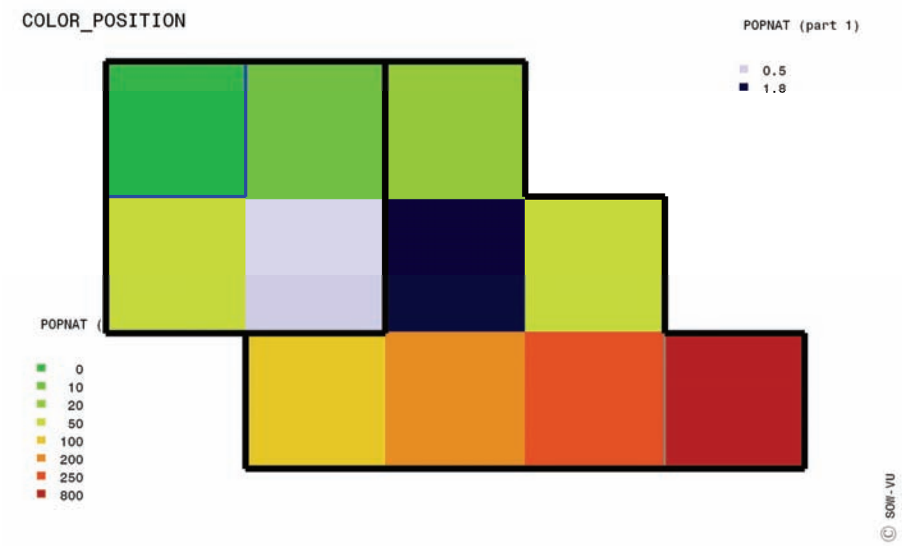

mapdat.locat_R
R
Region_ten
Region_twenty
© SOW-VU


POP_POP
POP
no data
0
10
20
50
100
200
250
800
© SOW-VU


POP_CN_POP
POP
0
60
620
800
© SOW-VU

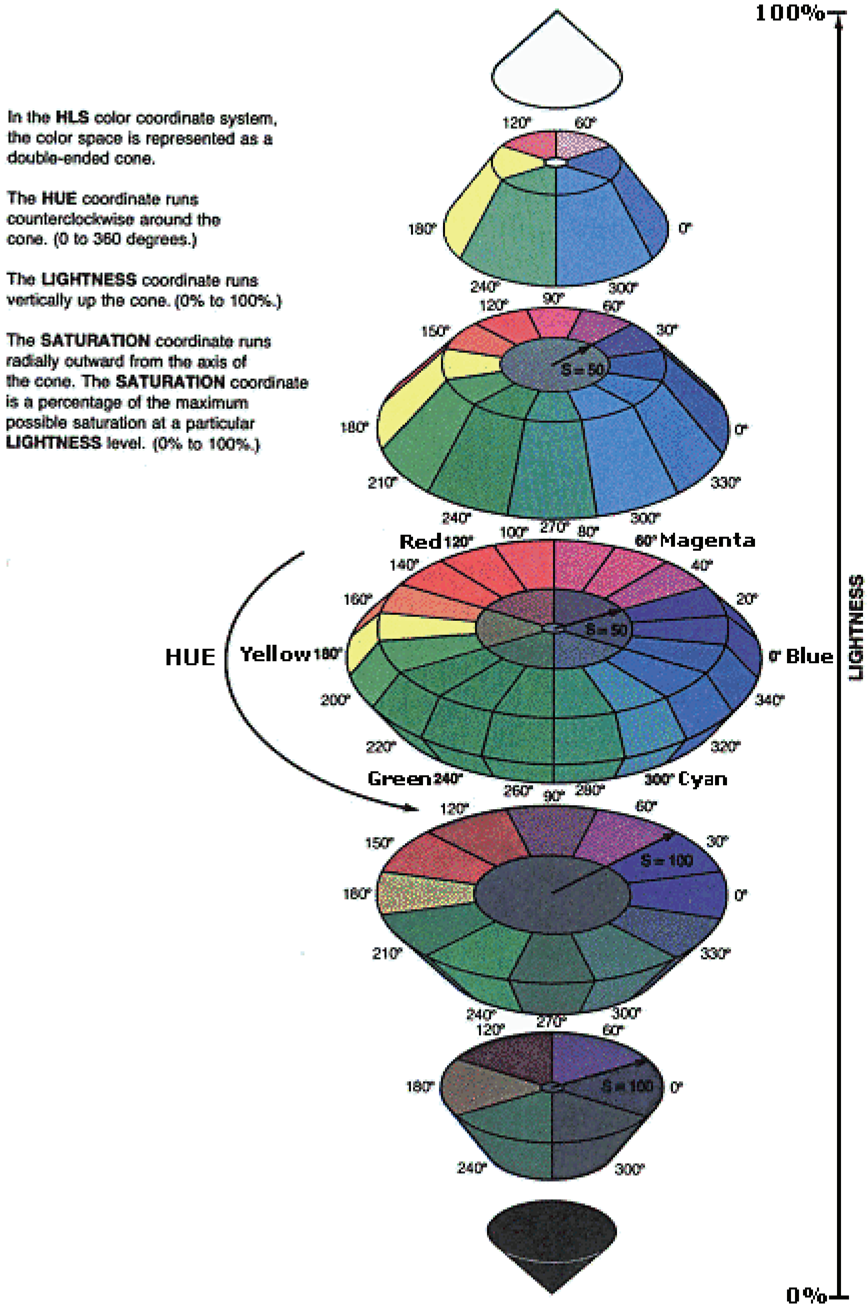

POP_NAT_POP
NAT
0.5
1.8
POP
0
10
20
50
100
200
250
800
© SOW-VU


POPNAT_POPNAT
POPNAT (part 2)
0
10
20
50
100
200
250
800
POPNAT (part 1)
0.5
1.8
© SOW-VU


COLOR_POSITION
POPNAT (part 1)
0.5
1.8
POPNAT (
0
10
20
50
100
200
250
800
© SOW-VU

Annex 2. Color schemes of the facility

The color schemes available for plotting have been selected using "Specifying Colors in SAS/GRAPH Programs". The colors are based on the double-ended cone depicted below.



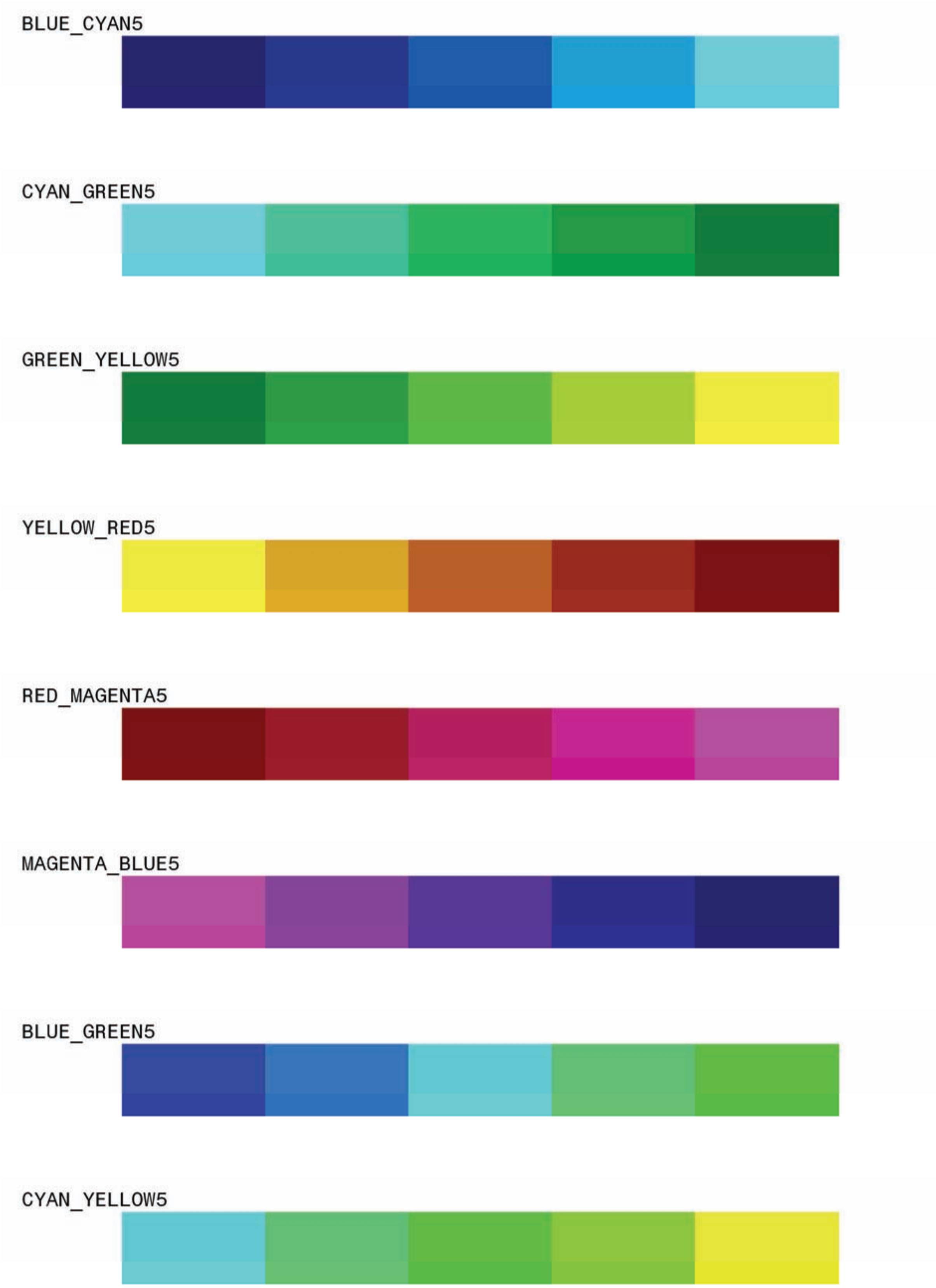
From this cone we defined a total of 60 color schemes. The default scheme is defined from the colors green and red. In terms of the cone depicted, the scheme starting at HUE coordinate 240° (green) and ends at HUE coordinate 120° (red). Colors in between are defined at equidistant points, following the number of colors required. The colors are given a standard lightness of 50%, except at the tails where lightness is reduced to 40% in order to improve appearance. The default scheme is named color = `GREEN_RED_` after its start and end color with the lower bar indicating reduced lightness at the tails. For a plot that uses 5 and 10 colors, respectively, it looks as follows.

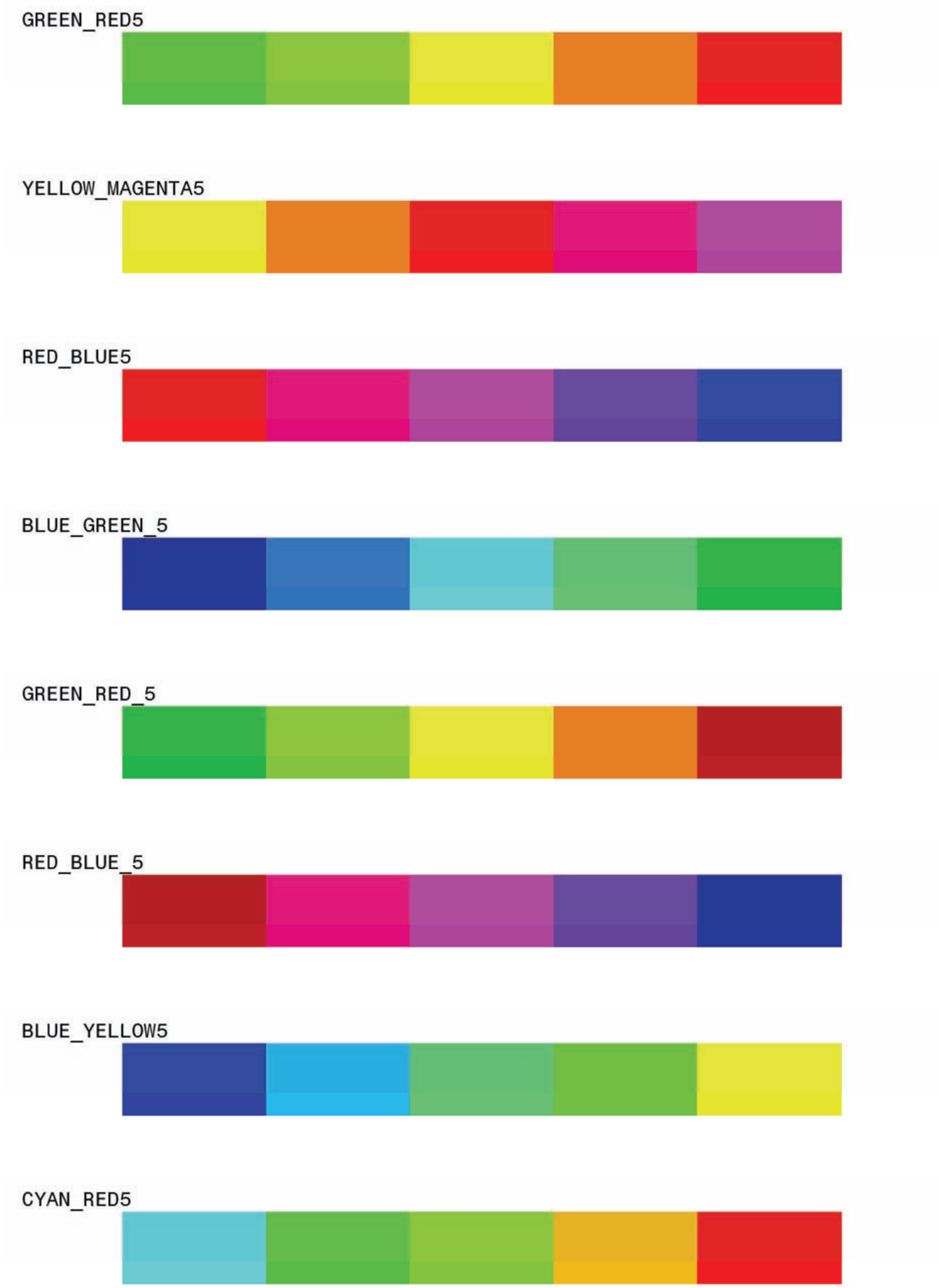
If two variables are plotted on a single map, the color scheme of the second variable will be `color2 = BLUE` starting and ending at HUE coordinate 0^o with lightness gradually decreasing from 90.5 to 12.5%.

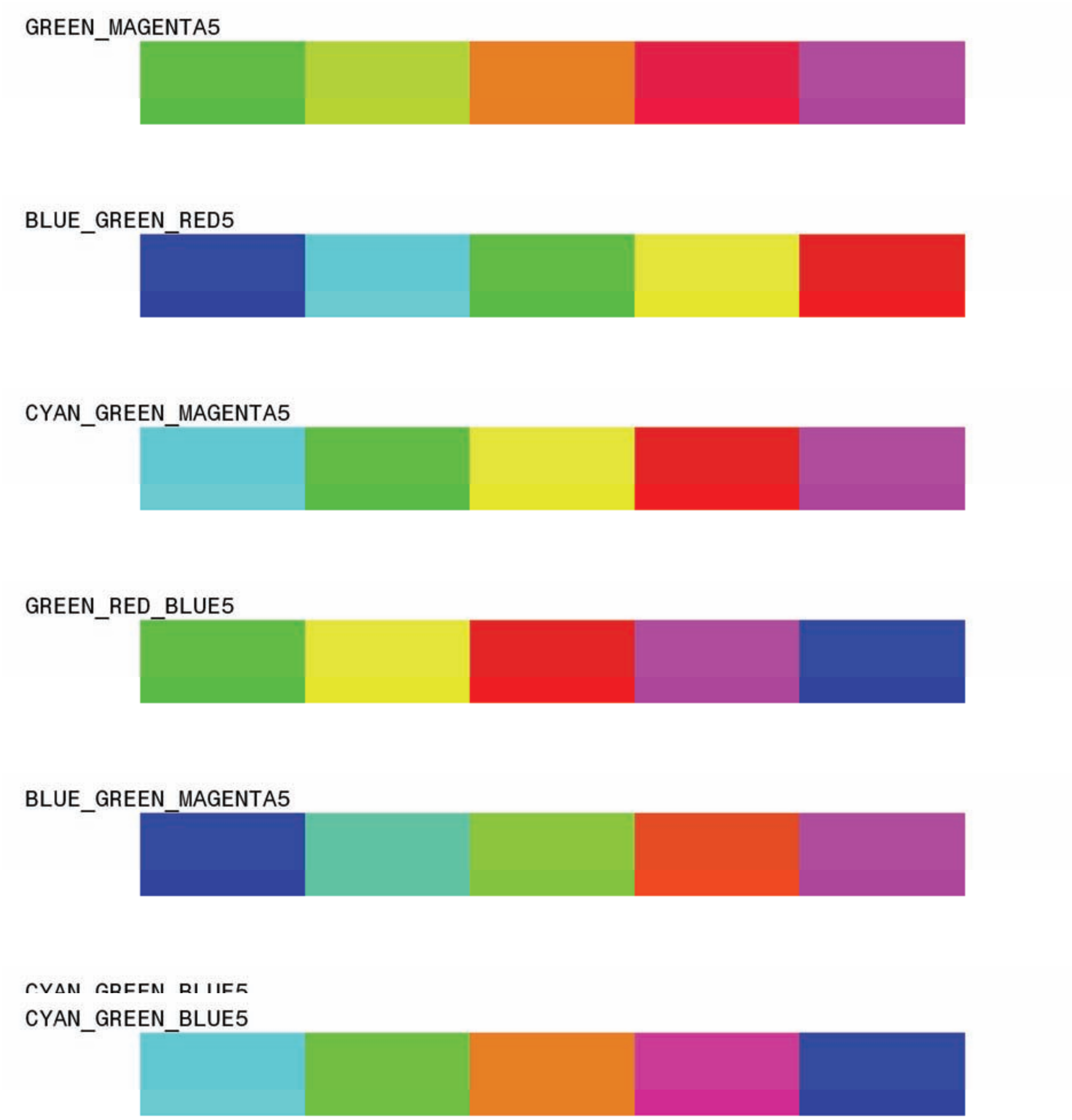
The total of 60 schemes available in the facility, all named after their start and end color, are listed below. A sample of half of the schemes is provided and the other half is just the reverse that follows from interchanging the end and the begin color.

```
(PLAIN COLORS FROM LIGHT TO DARK AND FROM DARK TO LIGHT)
BLUE               BLUE_R
MAGENTA            MAGENTA_R
RED                RED_R
YELLOW             YELLOW_R
GREEN              GREEN_R
CYAN               CYAN_R

(SIX COLORS TOGETHER)
SPECTRUM           SPECTRUM_R

(GRAY SHADES FROM LIGHT TO DARK AND FROM DARK TO LIGHT)
GRAY               GRAY_R

(MIX OF TWO NEARBY COLORS SPANNING 60 DEGREES)
BLUE_CYAN          CYAN_BLUE
CYAN_GREEN         GREEN_CYAN
GREEN_YELLOW       YELLOW_GREEN
YELLOW_RED         RED_YELLOW
RED_MAGENTA        MAGENTA_RED
MAGENTA_BLUE       BLUE_MAGENTA

(MIX OF TWO COLORS SPANNING 120 DEGREES)
BLUE_GREEN         GREEN_BLUE
CYAN_YELLOW        YELLOW_CYAN
GREEN_RED          RED_GREEN
YELLOW_MAGENTA     MAGENTA_YELLOW
RED_BLUE           BLUE_RED

(MIX OF TWO COLORS WITH DARKENED TAILS SPANNING 120 DEGREES)
BLUE_GREEN_        GREEN_BLUE_
GREEN_RED_         RED_GREEN_
RED_BLUE_          BLUE_RED_

(MIX OF TWO OPPOSITE COLORS SPANNING 180 DEGREES)
BLUE_YELLOW        YELLOW_BLUE
CYAN_RED           RED_CYAN
GREEN_MAGENTA      MAGENTA_GREEN

(SCHEMES OF THREE COLORS SPANNING 240 DEGREES)
BLUE_GREEN_RED     RED_GREEN_BLUE
CYAN_GREEN_MAGENTA MAGENTA_GREEN_CYAN
GREEN_RED_BLUE     BLUE_RED_GREEN

(SCHEMES OF THREE COLORS SPANNING 300 DEGREES)
BLUE_GREEN_MAGENTA MAGENTA_GREEN_BLUE
CYAN_GREEN_BLUE    BLUE_GREEN_CYAN
```


BLUE5
MAGENTA5
RED5
YELLOW5
GREEN5
CYAN5
SPECTRUM5
GRAY5


BLUE_CYAN5
CYAN_GREEN5
GREEN_YELLOW5
YELLOW_RED5
RED_MAGENTA5
MAGENTA_BLUE5
BLUE_GREEN5
CYAN_YELLOW5


GREEN_RED5
YELLOW_MAGENTA5
RED_BLUE5
BLUE_GREEN_5
GREEN_RED_5
RED_BLUE_5
BLUE_YELLOW5
CYAN_RED5


GREEN_MAGENTA5
BLUE_GREEN_RED5
CYAN_GREEN_MAGENTA5
GREEN_RED_BLUE5
BLUE_GREEN_MAGENTA5
CYAN_GREEN_BLUE5

Annex 3. List of options of the plotting facility

```
MINIMUM REQUIRED PARAMETERS
Plotting raster map with options set at their default
%map_grid
  (dataset = [specify SAS-dataset with LAT LON identifier]
  ,variable= [choose a variable from the dataset]
  ) ;

Plotting administrative map with options set at their default
%map_CN_PV_R
  (dataset = [specify SAS-dataset with CN, PV, R or ML identifier]
  ,variable= [choose a variable from the dataset]
  ,CN_PV_R = [specify aggregation level; must be CN, PV, R or ML]
  ) ;

Plotting raster map with two variables
%map_grid
  (dataset  = [specify SAS-dataset with LAT LON identifier]
  ,variable = [choose a first variable from the dataset]
  ,variable2= [choose a second variable from the dataset]
  ) ;

Plotting administrative map with two variables
%map_CN_PV_R
  (dataset = [specify SAS-dataset with CN, PV, R or ML identifier]
  ,variable= [choose a variable from the dataset]
  ,variable2= [choose a second variable from the dataset]
  ,CN_PV_R = [specify aggregation level; must be CN, PV, R or ML]
  ) ;

Plotting raster map of combi-variable with two ranges
%macro map_grid_combi
  (dataset    = [specify SAS-dataset with LAT LON identifier]
  , combi_var  = [choose the combi-variable from the dataset]
  ,first      = [threshold value between first and second part]
  ) ;
```

```
OPTIONS SPECIFIC FOR %map_grid and %map_grid_combi
  backgrnd     labels LAT LON not in dataset as 'backgrnd'  (default: =no data)

COMMON OPTIONS OF %map_grid, %map_CN_PV_R and %map_grid_combi
  ,device       graphic device                              (default: =gif_win2)
  ,graph        name of output file                         (default: =<dataset>_<variable>)
  ,dpi          resolution (dots per inch)                  (default: =300, maximum 800)
  ,scale_gr     scales down graph (factor)                  (default: =0.95, maximum 1)
  ,height_lgd   height of legend                            (default: =2)

  ,text1        first line of the title                     (default: =<dataset>_<variable>)
  ,text2        second line of the title                    (empty default)
  ,text3        third line of the title                     (empty default)
  ,text4        fourth line of the title                    (empty default)

  ,anno        additional annotate maps               (empty default, standard annotate is Anno_CN Anno_PV Anno_R)
  ,scale_ls     scales linesize of annotate map             (default: =1, =0 removes standard annotate)

  ,CN_sel       selection from CN codes                     (empty default, use as CN_sel=(1:5,82) )
  ,PV_sel       selection from PV codes                     (empty default, use as PV_sel=(2,7) )
  ,R_sel        selection from R codes                      (empty default, use as R_sel=(1,4) )
  ,ML_sel       selection from ML codes                     (empty default, use as ML_sel=(-1,0,1) )
  ,gmap_gplot   use gmap or gplot procedure                 (default: =gmap, ONLY USED IN CASE OF 1 variable plot)
  ,scale_gplot  scales the plotting square (factor)         (default: =1, relevant only for gplot procedure)
  ,enlarge      use entire plotting area for selection     (default: =yes, =no only possible with gplot procedure)
  ,format       classification combi-variable               (default: =EQUIPERC for both parts)
  ,levels       number of classes first part                (default: =5)
  ,decim        format in legend first part                 (default based on data range)
  ,color        color scheme first part                     (default: =GREEN_RED_)
  ,position     position of legend first part               (default: =1)
  ,label        label for first part                        (default: =<variable> (part 1))

  ,levels2      number of classes second part               (default: 5)
  ,decim2       format in legend second part                (default based on data range)
  ,color2       color scheme second part                    (default: =BLUE)
  ,position2    position of legend second part              (default: =3)
  ,label2       label for second part                       (default: <variable> (part 2))
```

```
DETAILS

FORMAT   FORMAT2
  =EQUIPERC (values will be classified into quantiles, see 'LEVELS' for number of quantiles <=10)]
  =EQUIDIST (values will be classified into ranges of equal length, see 'LEVELS' for number of intervals <=10)
  =DISCRETE (values are checked for discrete number of levels; must be <=10, otherwise FORMAT=EQUIPERC is taken)
  =[ SAS format, excluding the dot ] (the specified format will be used and 'LEVELS' and 'DECIM' are ignored)

LEVELS   LEVELS2
  =[ number between 1 and 10 ]

DECIM    DECIM2
  =12.3 if range/LEVELS<0.1
  =12.2 if 0.1<=range/LEVELS<1
  =12.1 if 1<=range/LEVELS<10
  =12.0 if 10<=range/LEVELS

COLOR    COLOR2
  =[ name of color scheme, see Annex 2 ]

POSITION POSITION2
  =[ number between 1 and 8, one of eight wind directions ]
   6 7 8
   4 * 5
   1 2 3

LABEL    LABEL2
  =[ text string of up to 40 characters ]

DEVICE
  =gif_win2 or =win to plot on the screen

GRAPH
  =[ text string of up to 40 characters ]

DPI
  =[ number between 100 and 600 ]
```

```
SCALE_GR
  =[ number between 0.1 and 1 ]

HEIGHT_LGD
  =[ number between 0.1 and 10 ]

TEXT1    TEXT2    TEXT3    TEXT4
  =[ text strings of up to 40 characters ]

ANNO
  =[ SAS annotate data set ]

SCALE_LS
  =[ number between 0 and 1; =0 removes standard annotate ]

CN_sel   PV_sel   R_sel    ML_sel
  =[ series of codes to be selected ]
  e.g CN_sel =(1:5,82)  selects CN codes 1, 2, 3, 4, 5 and 82
      PV_sel =(2,7)     selects PV codes 2 and 7
      R_sel =(104)      selects R code 104
      ML_sel=(-1,0,1) ) selects ML codes -1, 0 and 1

BACKGRND
  =[ text string of up to 40 characters ]

GMAP_GPLOT
  =gmap   a single variable is normally plotted with procedure gmap (fills cells nicely)
          two variables are automatically plotted on a single map with procedure gplot (not possible with gmap)!!!
  =gplot  when plot of single variable needs exact matching with plot of two variables

SCALE_GPLOT
  =[ number close to 1 ] if procedure gplot is used the plotting symbol (a square) might need to be scaled

ENLARGE
  =YES the map of selected CN PV R ML codes is enlarged so as to make use of the entire plotting area
  =NO   the map of selected codes uses plotting area of the master map (requires use of gplot procedure)
```

The Centre for World Food Studies (Dutch acronym SOW-VU) is a research institute related to the Department of Economics and Econometrics of the Vrije Universiteit Amsterdam. It was established in 1977 and engages in quantitative analyses to support national and international policy formulation in the areas of food, agriculture and development cooperation.

SOW-VU's research is directed towards the theoretical and empirical assessment of the mechanisms which determine food production, food consumption and nutritional status. Its main activities concern the design and application of regional and national models which put special emphasis on the food and agricultural sector. An analysis of the behaviour and options of socio-economic groups, including their response to price and investment policies and to externally induced changes, can contribute to the evaluation of alternative development strategies.

SOW-VU emphasizes the need to collaborate with local researchers and policy makers and to increase their planning capacity.

SOW-VU's research record consists of a series of staff working papers (for mainly internal use), research memoranda (refereed) and research reports (refereed, prepared through team work).

Centre for World Food Studies
SOW-VU
De Boelelaan 1105
1081 HV Amsterdam
The Netherlands

Telephone (31) 20 – 598 9321
Telefax (31) 20 – 598 9325
Email pm@sow.vu.nl
www http://www.sow.vu.nl/