

The 'rasterfile' format

Robert J. Hijmans

March 11, 2013

1 Introduction

The **'raster'** package has a default 'native' file format called 'rasterfile'. This file format is used because it is simple, flexible and extensible and does not require rgdal, which may be difficult to install (the **'raster'** package can read and write other formats via the **'rgdal'** package).

The rasterfile format is highly similar to many other formats used for raster data. It consists of two files. One file with sequential binary values (filename extension is '.gri'), and one header file (filename extension is '.grd'). The variation between such file formats is in the header file, and the contents of the rasterfile header file are described here.

The purpose is to standardize the format and help others to read and write files of the same format if they wish to do so. This vignette is aimed at software developers. The typical user of the **'raster'** package does not need to be familiar with it.

This is an incomplete first draft.

2 ini files

The header ('.grd') file is organized as a windows '.ini' file. This is a simple database format that is subdivided in sections. Within each section there are variables and their values.

Thus, .ini files have a layout like this:

```
[section1]
var1=value
var2=value
[section2]
var3=value
var4=value
```

Variables names must be unique within a section, but the same variable name could occur in multiple sections. This is not done for raster files (variable names are unique) such that section names could be ignored. The **'raster'** package has a convenient function, **readIniFile**, to read .ini files.

3 Sections

3.1 general

This section has two variables, 'creator' and 'created'. For example:

```
[general]
creator=R package 'raster'
created= 2010-03-13 17:26:34
Neither of these variables is required.
```

3.2 georeference

This section has the number of rows (nrows) and columns (ncols), and describes the spatial extent (bounding box) with four variables (xmin, xmax, ymin, ymax). These variables are required. The number of rows and columns are integers ≥ 1 . The extent variables are numeric with $xmin < xmax$ and $ymin < ymax$. The coordinates refer to the extremes of the outer cells (not to the centers of these cells). Resolution (cell size) is not specified (it is derived value from the extent and the number of columns and rows).

An additional optional variable 'projection' has the coordinate reference system in PROJ4 syntax.

```
[georeference]
nrows= 100
ncols= 100
xmin= -180
ymin= -90
xmax= 180
ymax= 90
projection= +proj=longlat +datum=WGS84
```

3.3 data

This subsection has information about the file type as well as the cell values. Here is an example

```
[data]
datatype= FLT4S
nodatavalue= -3.4e+38
byteorder= little
nbands= 1
bandorder= BIL
categorical= FALSE
levels=
minvalue= 1
maxvalue= 10000
```

datatype is required. Its values must be one of 'LOG1S', 'INT1S', 'INT2S', 'INT4S', 'INT8S', 'INT1U', 'INT2U', 'FLT4S', 'FLT8S'. The first three letters

indicate the type of number that is stored (logical, integer, or float). The fourth character determines how many bytes are used to store these. The last letter indicates, if applicable, whether the values are signed or not (i.e. whether negative values are possible).

`nodatavalue` is optional (but necessary if there are nodata (NA) values). It can be any value. But in the raster package the lowest possible value is used for signed integer and float data types and the highest integer is used for unsigned integer types (this is to avoid using 0 as the nodata value).

`byteorder` is optional but recommended. It should be either 'big' (linux) or 'little' (windows). If absent, the raster package assumes the platform byte order.

`nbands` is optional. It indicates the number of layers (bands) stored in the file and hence its values should be an integer ≥ 1 . If absent, the raster package assumes it is 1.

`bandorder` is required if `nbands` > 1 and ignored when `nbands`=1. Values can be 'BIL' (band interleaved by line), 'BIP' (band interleaved by pixel) and 'BSQ' (band sequential). BIL is recommended for most cases.