Mapping with R

Ernest Guevarra

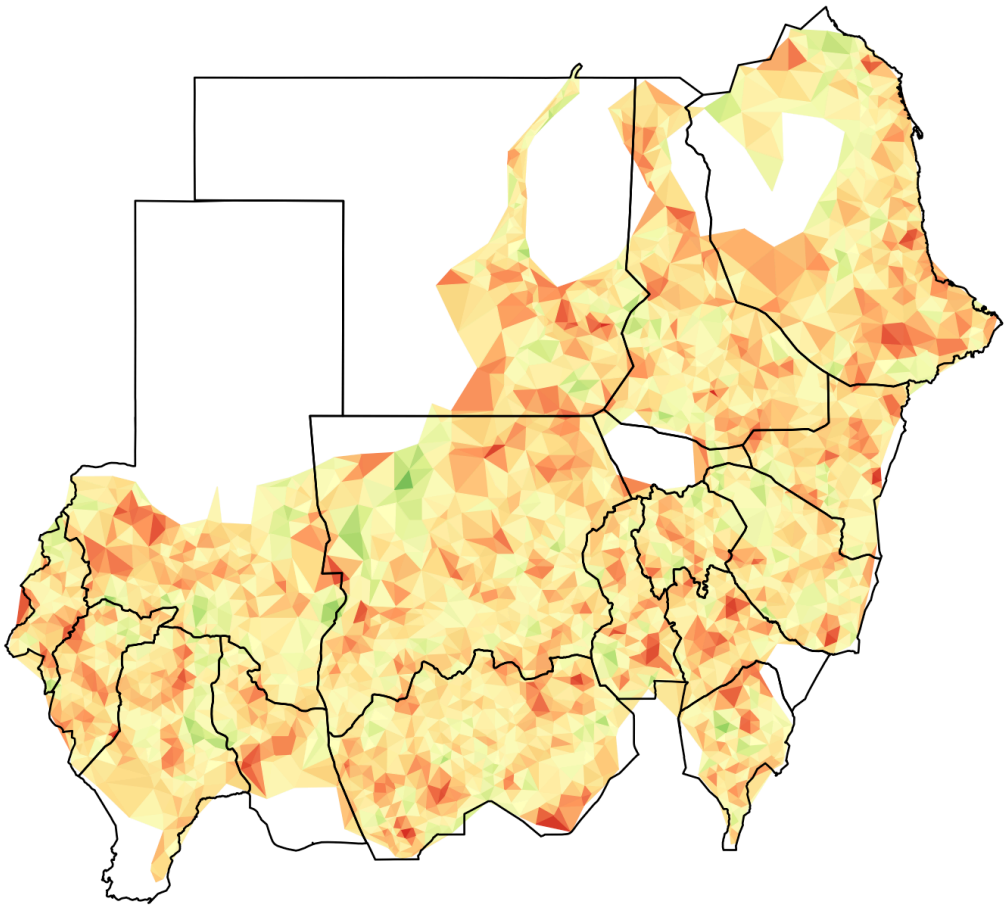
2018-10-07

Contents

List of Tables

List of Figures

Short course on the use of R for the mapping requirements of S3M



1 Retrieving map data in R

In this exercise we will use **R** to read a **shapefile** dataset and get oriented with the structure and features of a **shapefile** dataset. The aim of the exercise is for you to become familiar with the use of R in handling **shapefile** datasets.

By this time, you have already learned how to issue a command to retrieve a standard or typical dataset using the `read.table()` function

For this exercise, we will use the `readOGR()` function provided by the `rgdal` package to retrieve **shapefile** dateset.

First, we need to install and load the `rgdal` package.

```
install.packages("rgdal")
library(rgdal)
```

We can now try to read the Sudan **shapefile**. To do this however, we need to have an orientation on what **shapefiles** are.

A **shapefile** is a digital vector storage format for storing geometric location and associated attribute information. This format lacks the capacity to store topological information. The **shapefile** format was initially developed for proprietary use with ArcView GIS version 2 in the early 1990s. It is now possible to read and write **shapefiles** using a variety of programs including data analysis software such as **R**.

Shapefiles are simple because they store the primitive geometric data types of points, lines, and polygons. They are of limited use without any attributes to specify what they represent. Therefore, a table of records will store properties/attributes for each primitive shape in the **shapefile**. Shapes (points/lines/polygons) together with data attributes can create infinitely many representations about geographic data. Representation provides the ability for powerful and accurate computations.

While the term "shapefile" is quite common, a **shapefile** is actually a set of several files. Three individual files are mandatory to store the core data that comprise a **shapefile**:

- .shp
- .shx
- .dbf

The actual **shapefile** relates specifically to `.shp` files but alone is incomplete for distribution, as the other supporting files are required.

With this knowledge of **shapefiles**, let us now take a look at the Sudan **shapefiles** dataset.

The Sudan **shapefiles** dataset contains the following **shapefiles**:

Table 1.1: Sudan shapefiles structure

Directory name	Directory files	Description
sudan01	sudan01.shp	Polygon shapefile of Sudan up to state administrative level
	sudan01.shx	
	sudan01.dbf	
	sudan01.prj	
	sudan01.qpj	
sudan02	sudan02.shp	Polygon shapefile of Sudan up to locality administrative level
	sudan02.shx	
	sudan02.dbf	
	sudan02.prj	
	sudan02.qpj	
grid12poly	grid12poly.shp	Polygon shapefile of rectangular grid at d = 12km
	grid12poly.shx	
	grid12poly.dbf	
	grid12poly.prj	
	grid12poly.qpj	
grid12kmSudan	grid12kmSudan.shp	Line shapefile of rectangular grid at d = 12km
	grid12kmSudan.shx	
	grid12kmSudan.dbf	
	grid12kmSudan.prj	
	grid12kmSudan.qpj	

We can now retrieve these shapefile datasets and create an object for each one using the `readOGR()` function from the `rgdal` package.

```
sudan01 <- readOGR(dsn = "maps", layer = "sudan01")
#> OGR data source with driver: ESRI Shapefile
#> Source: "/Users/ernest/Documents/GitHub/map-r/maps", layer: "sudan01"
#> with 18 features
```

```
#> It has 8 fields

sudan02 <- readOGR(dsn = "maps", layer = "sudan02")
#> OGR data source with driver: ESRI Shapefile
#> Source: "/Users/ernest/Documents/GitHub/map-r/maps", layer: "sudan02"
#> with 169 features
#> It has 15 fields

grid12poly <- readOGR(dsn = "maps", layer = "grid12poly")
#> OGR data source with driver: ESRI Shapefile
#> Source: "/Users/ernest/Documents/GitHub/map-r/maps", layer: "grid12poly"
#> with 13950 features
#> It has 5 fields
#> Integer64 fields read as strings:  ID

grid12kmSudan <- readOGR(dsn = "maps", layer = "grid12kmSudan")
#> OGR data source with driver: ESRI Shapefile
#> Source: "/Users/ernest/Documents/GitHub/map-r/maps", layer: "grid12kmSudan"
#> with 245 features
#> It has 2 fields
#> Integer64 fields read as strings:  ID
```

This series of commands illustrates key things about the way shapefile data can be read and handled in **R**.

First is that the retrieval of **shapefile** datasets follows a very similar syntax as that of other standard datasets but just using the `readOGR()` function. The same principles apply including ensuring that you specify the corresponding directory in which your **shapefiles** are stored.

Now that we have stored the various shapefile data into objects, we can now explore and get ourselves oriented to the structure and features of a **shapefile** data object. We will do this using standard / basic functions in **R** that you have learned already in the previous training.

First, let us learn the class of a **shapefile** data object. We can find this out using the same function that you are familiar with already and have used previously the `class()` function.

```
class(sudan01)
```

This command gives the following output:

```
#> [1] "SpatialPolygonsDataFrame"
#> attr(,"package")
#> [1] "sp"
```

This tells us that the `sudan01` object is of class `SpatialPolygonsDataFrame`. It also tells us that this is a special class specific to the `sp` package.

The `sp` package provides classes and methods for spatial data. The classes document where the spatial location information resides, for 2D or 3D data. Utility functions are provided, e.g. for plotting data as maps, spatial selection, as well as methods for retrieving coordinates, for subsetting, print, summary, etc.

If you check for the class of the other **shapefile** objects you've created, you will see that all of them are of the same `SpatialPolygonsDataFrame` class except for `grid12kmSudan`. Checking for the class of `grid12kmSudan` revealed the following:

```
class(grid12kmSudan)
#> [1] "SpatialLinesDataFrame"
#> attr(,"package")
#> [1] "sp"
```

This tells us that the `grid12kmSudan` objects is of class `SpatialLinesDataFrame`.

You are now getting introduced to two of the most common shapes of a **shapefile**: *polygons* and *lines*.

A *polygon* consists of one or more rings. A ring is a connected sequence of four or more points that form a closed, non-self-intersecting loop. A *polygon* may contain multiple outer rings. The order of vertices or orientation for a ring indicates which side of the ring is the interior of the *polygon*. The neighbourhood to the right of an observer walking along the ring in vertex order is the neighbourhood inside the *polygon*. Vertices of rings defining holes in *polygons* are in a counterclockwise direction. Vertices for a single, ringed *polygon* are, therefore, always in clockwise order. The rings of a *polygon* are referred to as its parts.

A *line* is an ordered set of vertices that consists of one or more parts. A part is a connected sequence of two or more points. Parts may or may not be connected to one another. Parts may or may not intersect one another.

One of the other shapes that **shapefiles** take or represent is *points*.

A *point* consists of a pair of double-precision coordinates in the order `x`, `y`.

Because of this simple property of a *point* **shapefile** (i.e. a basic set of `x` and `y` coordinates), the use of **shapefile** format to store the *point* shape is not commonly used. The `x` and `y` coordinates for *points* can be contained or stored in other more basic formats such as CSV.

For example, the dataset that contains the `x` and `y` coordinates of all the known villages in Sudan is named `settlementsSudan.csv`. If we create an object called villages for this dataset

```
villages <- read.csv("maps/settlementsSudan.csv", header = TRUE, sep = ",")
```

and use the `head()` function to view the first 10 rows of this dataset

```
head(villages, 10)
```

we get:

```
#>    ID      Village Pop     Source      State Locality        X        Y
#> 1   1        Kosti         Georef White Nile    Kosti 32.66751 13.14846
#> 2   2     Tandalti     Calculated White Nile    Kosti 31.86393 13.00969
#> 3   3    Qawz kobi            GPS White Nile    Kosti 32.35000 13.60000
#> 4   4    Karjuggle            GPS White Nile    Kosti 32.38333 13.46667
#> 5   5   Idd maktuf            GPS White Nile    Kosti 32.28333 13.50000
#> 6   6       Maryam            GPS White Nile    Kosti 32.55000 13.46667
#> 7   7 Qawz nyaneir            GPS White Nile    Kosti 32.28333 13.60000
#> 8   8       Salogi            GPS White Nile    Kosti 32.45000 13.15000
#> 9   9      Sulayah            GPS White Nile    Kosti 32.25000 13.26667
#> 10 10      Seleima     Calculated White Nile    Kosti 32.05833 13.00833
#>    Remarks
#> 1
#> 2
#> 3
#> 4
#> 5
#> 6
#> 7
#> 8
#> 9
#> 10
```

As you will notice here, the villages object contains information on the `x` and `y` coordinates of each of the villages in Sudan. So, whilst this dataset is not a **shapefile** (it is a basic data frame), it has information and a structure that is comparable to a point **shapefile** as defined above.

In the succeeding exercises, this similarity of a *point* **shapefile** and a standard data frame containing `x` and `y` coordinates of *points* will be further discussed and illuminated.

This knowledge on classes and shapes of **shapefiles** is an important learning particularly when performing functions to handle or manipulate different **shapefile** objects. The general principle is that functions or operations between two or more **shapefile** objects require these objects to be of the same class or family of classes. Also, the shapes defined by the **shapefile** object determine the way the **shapefile** data is structured which in turn determine how these objects can and should be handled or manipulated in **R**. These principles will be further illuminated in the succeeding exercises.

After learning about the class of **shapefile** objects, we now learn about the structure of these objects. We are able to appreciate the structure of a **shapefile** object by using the function `str()`.

```
str(sudan01)
```

The output of this command is:

```
#> Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots
#>   ..@ data       :'data.frame':  18 obs. of  8 variables:
#>   .. ..$ State     : chr [1:18] "West Darfur" "Central Darfur" "North Darfur" "East D
#>   .. ..$ Old_state : chr [1:18] "West Darfur" "West Darfur" "North Darfur" "South Dar
#>   .. ..$ New_State : chr [1:18] "West Darfur" "Central Darfur" "North Darfur" "East D
#>   .. ..$ STATE_1   : chr [1:18] NA NA NA NA ...
#>   .. ..$ STATEAR   : chr [1:18] NA NA NA NA ...
#>   .. ..$ Source    : chr [1:18] NA NA NA NA ...
#>   .. ..$ STATE_CODE: chr [1:18] NA NA NA NA ...
#>   .. ..$ ISO_CODE  : chr [1:18] NA NA NA NA ...
#>   ..@ polygons   :List of 18
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
```

```
#>   .. .. .. .. .. .. ..@ labpt  : num [1:2] 22.6 13.5
#>   .. .. .. .. .. .. ..@ area   : num 1.86
#>   .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. ..@ coords : num [1:981, 1:2] 22.1 22.1 22.1 22.2 22.2 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 22.6 13.5
#>   .. .. .. ..@ ID       : chr "0"
#>   .. .. .. ..@ area     : num 1.86
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. ..@ labpt  : num [1:2] 23.4 12.3
#>   .. .. .. .. .. .. ..@ area   : num 2.75
#>   .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. ..@ coords : num [1:463, 1:2] 23.6 23.6 23.6 23.6 23.7 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 23.4 12.3
#>   .. .. .. ..@ ID       : chr "1"
#>   .. .. .. ..@ area     : num 2.75
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. ..@ labpt  : num [1:2] 25.6 16.2
#>   .. .. .. .. .. .. ..@ area   : num 26.7
#>   .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. ..@ coords : num [1:669, 1:2] 27.5 27.5 27.2 27.1 26.9 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 25.6 16.2
#>   .. .. .. ..@ ID       : chr "2"
#>   .. .. .. ..@ area     : num 26.7
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. ..@ labpt  : num [1:2] 26.5 11
#>   .. .. .. .. .. .. ..@ area   : num 4.44
#>   .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. ..@ ringDir: int 1
```

```
#>   .. .. .. .. .. .. .. ..@ coords : num [1:932, 1:2] 25.5 25.6 25.6 25.6 25.6 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 26.5 11
#>   .. .. .. ..@ ID       : chr "3"
#>   .. .. .. ..@ area     : num 4.44
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 24.4 11
#>   .. .. .. .. .. .. .. ..@ area   : num 6.91
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:1896, 1:2] 24.4 24.4 24.4 24.4 24.4 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 24.4 11
#>   .. .. .. ..@ ID       : chr "4"
#>   .. .. .. ..@ area     : num 6.91
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 33.3 14.6
#>   .. .. .. .. .. .. .. ..@ area   : num 2.28
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:490, 1:2] 33.6 33.6 33.6 33.6 33.6 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 33.3 14.6
#>   .. .. .. ..@ ID       : chr "5"
#>   .. .. .. ..@ area     : num 2.28
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 34.1 11.3
#>   .. .. .. .. .. .. .. ..@ area   : num 3.16
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:406, 1:2] 34.5 34.5 34.5 34.5 34.5 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 34.1 11.3
#>   .. .. .. ..@ ID       : chr "6"
```

```
#>   .. .. .. ..@ area     : num 3.16
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 35.1 14.2
#>   .. .. .. .. .. .. .. ..@ area   : num 4.99
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:505, 1:2] 34.2 34.3 34.4 34.4 34.5 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 35.1 14.2
#>   .. .. .. ..@ ID       : chr "7"
#>   .. .. .. ..@ area     : num 4.99
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 35.9 15.8
#>   .. .. .. .. .. .. .. ..@ area   : num 4.1
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:294, 1:2] 35.7 36 36.1 36.2 36.2 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 35.9 15.8
#>   .. .. .. ..@ ID       : chr "8"
#>   .. .. .. ..@ area     : num 4.1
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 32.8 15.9
#>   .. .. .. .. .. .. .. ..@ area   : num 1.79
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:198, 1:2] 33.5 33.7 34 34 34 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 32.8 15.9
#>   .. .. .. ..@ ID       : chr "9"
#>   .. .. .. ..@ area     : num 1.79
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
```

```
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 33.5 18.3
#>   .. .. .. .. .. .. .. ..@ area   : num 11.1
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:226, 1:2] 35.6 35.4 35.4 35.3 35.3 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 33.5 18.3
#>   .. .. .. ..@ ID       : chr "10"
#>   .. .. .. ..@ area     : num 11.1
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 29.3 19.6
#>   .. .. .. .. .. .. .. ..@ area   : num 31.4
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:117, 1:2] 31.7 32.1 32.1 32.4 32.4 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 29.3 19.6
#>   .. .. .. ..@ ID       : chr "11"
#>   .. .. .. ..@ area     : num 31.4
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 35.7 19.8
#>   .. .. .. .. .. .. .. ..@ area   : num 18.6
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:2309, 1:2] 37 37 37 37 37 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 35.7 19.8
#>   .. .. .. ..@ ID       : chr "12"
#>   .. .. .. ..@ area     : num 18.6
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 2
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 33.2 11.5
#>   .. .. .. .. .. .. .. ..@ area   : num 0.00152
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
```

```
#>   .. .. .. .. .. .. .. ..@ coords : num [1:10, 1:2] 33.2 33.2 33.2 33.1 33.1 ...
#>   .. .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 34 12.9
#>   .. .. .. .. .. .. .. ..@ area   : num 3.27
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:545, 1:2] 33.9 33.9 33.9 33.9 33.9 ...
#>   .. .. .. ..@ plotOrder: int [1:2] 2 1
#>   .. .. .. ..@ labpt    : num [1:2] 34 12.9
#>   .. .. .. ..@ ID       : chr "13"
#>   .. .. .. ..@ area     : num 3.27
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 32.3 13.4
#>   .. .. .. .. .. .. .. ..@ area   : num 3.17
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:378, 1:2] 32.5 32.5 32.5 32.5 32.5 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 32.3 13.4
#>   .. .. .. ..@ ID       : chr "14"
#>   .. .. .. ..@ area     : num 3.17
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 30.8 11.3
#>   .. .. .. .. .. .. .. ..@ area   : num 6.49
#>   .. .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. .. ..@ coords : num [1:451, 1:2] 31.7 31.7 31.7 31.7 31.7 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 30.8 11.3
#>   .. .. .. ..@ ID       : chr "15"
#>   .. .. .. ..@ area     : num 6.49
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. .. ..@ labpt  : num [1:2] 28.4 11.8
#>   .. .. .. .. .. .. .. ..@ area   : num 9.47
```

```
#>   .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. ..@ coords : num [1:566, 1:2] 27.5 27.5 27.5 27.5 27.8 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 28.4 11.8
#>   .. .. .. ..@ ID       : chr "16"
#>   .. .. .. ..@ area     : num 9.47
#>   .. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. .. .. ..@ Polygons :List of 1
#>   .. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. .. .. ..@ labpt  : num [1:2] 29.7 14.8
#>   .. .. .. .. .. .. ..@ area   : num 15.7
#>   .. .. .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. .. .. ..@ coords : num [1:385, 1:2] 26.9 27.1 27.2 27.5 28.6 ...
#>   .. .. .. ..@ plotOrder: int 1
#>   .. .. .. ..@ labpt    : num [1:2] 29.7 14.8
#>   .. .. .. ..@ ID       : chr "17"
#>   .. .. .. ..@ area     : num 15.7
#>   ..@ plotOrder  : int [1:18] 12 3 13 18 11 17 5 16 8 4 ...
#>   ..@ bbox       : num [1:2, 1:2] 21.81 8.64 38.59 23.14
#>   .. ..- attr(*, "dimnames")=List of 2
#>   .. .. ..$ : chr [1:2] "x" "y"
#>   .. .. ..$ : chr [1:2] "min" "max"
#>   ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
#>   .. .. ..@ projargs: chr "+proj=utm +zone=36 +a=6378249.2 +b=6356514.999904194 +unit
```

which gives the class of the **shapefile** object and gives us an idea of the data structure as having 5 slots. This is one of the key differences of a `sp` data frame compared to a standard basic data frame. In a basic data frame, you basically have single data set organised in rows and columns similar to that of a table. In a `sp` data frame, you can think of it as a compound data frame in which each slot contains a specific dataset.

As you look further down into the structure of the **shapefile** object, you will notice the `@` symbol recurring 5 times. This is the symbol used to retrieve the different slots of the **shapefile** objects.

The 5 slots in a `SpatialPolygonsDataFrame` are:

Table 1.2: SpatialPolygonsDataFrame structure

data	Contains the index or reference data frame of the **shapefile** the number of rows of which indicates the number of polygons that comprise the entire **shapefile**.
polygons	Contains *n* number of datasets based on the number of *polygons* that comprise the entire **shapefile**.
plotOrder	Contains an integer vector with a length equal to the number of *polygons* that comprise the entire **shapefile** and have values starting from 1 to *n* number of *polygons* in the entire **shapefile**. The order of the values of this vector determines which *polygon* is drawn first when plotting. This order is determined by decreasing area size of the *polygons*.
bbox	Contains a matrix the values of which are the minimum and maximum `x` and `y` limits of the entire **shapefile**.
proj4string	Contains a character string that specifies the projection and datum of the **shapefile** object

Now let us try to extract the different slots of `sudan01` object. Making a call for the `data` slot gives:

```
sudan01@data
#>                State        Old_state        New_State STATE_1
#> 0        West Darfur       West Darfur       West Darfur    <NA>
#> 1     Central Darfur       West Darfur    Central Darfur    <NA>
#> 2       North Darfur      North Darfur      North Darfur    <NA>
#> 3        East Darfur      South Darfur       East Darfur    <NA>
#> 4       South Darfur      South Darfur      South Darfur    <NA>
#> 5          Al Gezira              <NA>              <NA>    <NA>
#> 6          Blue Nile              <NA>              <NA>    <NA>
#> 7            Gedaref              <NA>              <NA>    <NA>
#> 8            Kassala              <NA>              <NA>    <NA>
#> 9           Khartoum              <NA>              <NA>    <NA>
#> 10              Nile              <NA>              <NA>    <NA>
```

```
#> 11          Northern              <NA>              <NA>    <NA>
#> 12           Red Sea              <NA>              <NA>    <NA>
#> 13            Sennar              <NA>              <NA>    <NA>
#> 14        White Nile              <NA>              <NA>    <NA>
#> 15 Southern Kordofan Southern Kordofan Southern Kordofan    <NA>
#> 16  Western Kordofan    South Kordofan  Western Kordofan    <NA>
#> 17 Northern Kordofan Northern Kordofan Northern Kordofan    <NA>
#>         STATEAR Source STATE_CODE ISO_CODE
#> 0          <NA>   <NA>       <NA>     <NA>
#> 1          <NA>   <NA>       <NA>     <NA>
#> 2          <NA>   <NA>       <NA>     <NA>
#> 3          <NA>   <NA>       <NA>     <NA>
#> 4          <NA>   <NA>       <NA>     <NA>
#> 5       ÇáÌÒíÑÉ    CBS       SU04    SD-07
#> 6  Çáäíá ÇáÇÒÑÞ    CBS       SU02    SD-24
#> 7       ÇáÞÖÇÑÝ    CBS       SU05    SD-06
#> 8          ßÓáÇ    CBS       SU07    SD-05
#> 9       ÇáÎÑØæã    CBS       SU08    SD-03
#> 10    äåÑ Çäíá    CBS       SU10    SD-04
#> 11     ÇáÕãÇíÉ    CBS       SU11    SD-01
#> 12 ÇáÈÍÑ ÇáÇÍãÑ    CBS       SU15    SD-26
#> 13         ÚäÇÑ    CBS       SU16    SD-25
#> 14 Çäíá ÇáÇÈíÖ    CBS       SU25    SD-08
#> 15         <NA>   <NA>       <NA>     <NA>
#> 16         <NA>   <NA>       <NA>     <NA>
#> 17         <NA>   <NA>       <NA>     <NA>
```

Here we note that the `sudan01` **shapefile** contains *polygons* of the each of the states of Sudan with `sudan01@data` specifying the names of each of these states.

To check for the number of *polygons* in `sudan01` **shapefile** (the answer to which will also be the number of states in Sudan), we can use the `nrow()` function as follows:

```
nrow(sudan01@data)
```

which gives a result of

```
#> [1] 18
```

There are 18 polygons in `sudan01` **shapefile** which indicate that Sudan has about 18 states (if the **shapefile** used is up-to-date).

Let us now try extract the *polygons* slot of `sudan01`. If we make a call for the *polygons* slot as below:

```
sudan01@polygons
```

we end up with a very long output which is not easy to review or appreciate. This is understandable because from the previous command extracting the slot data of `sudan01` we know already that the *polygons* slot will have 18 sets of *polygon* **shapefile** data each of which will have it's own datasets and data structure. This means that the output will be very long and not easy to manage.

In this instance, a review of the structure of `sudan01@polygons` may help in getting an insight as to how to handle or manage the datasets contained inside this slot. We call the `str()` function on `sudan01@polygons` as follows:

```
str(sudan01@polygons)
#> List of 18
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 22.6 13.5
#>   .. .. .. .. ..@ area   : num 1.86
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:981, 1:2] 22.1 22.1 22.1 22.2 22.2 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 22.6 13.5
#>   .. ..@ ID       : chr "0"
#>   .. ..@ area     : num 1.86
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
```

```
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 23.4 12.3
#>   .. .. .. .. ..@ area   : num 2.75
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:463, 1:2] 23.6 23.6 23.6 23.6 23.7 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 23.4 12.3
#>   .. ..@ ID       : chr "1"
#>   .. ..@ area     : num 2.75
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 25.6 16.2
#>   .. .. .. .. ..@ area   : num 26.7
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:669, 1:2] 27.5 27.5 27.2 27.1 26.9 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 25.6 16.2
#>   .. ..@ ID       : chr "2"
#>   .. ..@ area     : num 26.7
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 26.5 11
#>   .. .. .. .. ..@ area   : num 4.44
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:932, 1:2] 25.5 25.6 25.6 25.6 25.6 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 26.5 11
#>   .. ..@ ID       : chr "3"
#>   .. ..@ area     : num 4.44
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 24.4 11
#>   .. .. .. .. ..@ area   : num 6.91
```

```
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:1896, 1:2] 24.4 24.4 24.4 24.4 24.4 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 24.4 11
#>   .. ..@ ID       : chr "4"
#>   .. ..@ area     : num 6.91
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 33.3 14.6
#>   .. .. .. .. ..@ area   : num 2.28
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:490, 1:2] 33.6 33.6 33.6 33.6 33.6 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 33.3 14.6
#>   .. ..@ ID       : chr "5"
#>   .. ..@ area     : num 2.28
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 34.1 11.3
#>   .. .. .. .. ..@ area   : num 3.16
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:406, 1:2] 34.5 34.5 34.5 34.5 34.5 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 34.1 11.3
#>   .. ..@ ID       : chr "6"
#>   .. ..@ area     : num 3.16
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 35.1 14.2
#>   .. .. .. .. ..@ area   : num 4.99
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:505, 1:2] 34.2 34.3 34.4 34.4 34.5 ...
#>   .. ..@ plotOrder: int 1
```

```
#>   .. ..@ labpt    : num [1:2] 35.1 14.2
#>   .. ..@ ID       : chr "7"
#>   .. ..@ area     : num 4.99
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 35.9 15.8
#>   .. .. .. .. ..@ area   : num 4.1
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:294, 1:2] 35.7 36 36.1 36.2 36.2 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 35.9 15.8
#>   .. ..@ ID       : chr "8"
#>   .. ..@ area     : num 4.1
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 32.8 15.9
#>   .. .. .. .. ..@ area   : num 1.79
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:198, 1:2] 33.5 33.7 34 34 34 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 32.8 15.9
#>   .. ..@ ID       : chr "9"
#>   .. ..@ area     : num 1.79
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 33.5 18.3
#>   .. .. .. .. ..@ area   : num 11.1
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:226, 1:2] 35.6 35.4 35.4 35.3 35.3 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 33.5 18.3
#>   .. ..@ ID       : chr "10"
#>   .. ..@ area     : num 11.1
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
```

```
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 29.3 19.6
#>   .. .. .. .. ..@ area   : num 31.4
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:117, 1:2] 31.7 32.1 32.1 32.4 32.4 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 29.3 19.6
#>   .. ..@ ID       : chr "11"
#>   .. ..@ area     : num 31.4
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 35.7 19.8
#>   .. .. .. .. ..@ area   : num 18.6
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:2309, 1:2] 37 37 37 37 37 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 35.7 19.8
#>   .. ..@ ID       : chr "12"
#>   .. ..@ area     : num 18.6
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 2
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 33.2 11.5
#>   .. .. .. .. ..@ area   : num 0.00152
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:10, 1:2] 33.2 33.2 33.2 33.1 33.1 ...
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 34 12.9
#>   .. .. .. .. ..@ area   : num 3.27
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:545, 1:2] 33.9 33.9 33.9 33.9 33.9 ...
#>   .. ..@ plotOrder: int [1:2] 2 1
#>   .. ..@ labpt    : num [1:2] 34 12.9
#>   .. ..@ ID       : chr "13"
```

```
#>   .. ..@ area     : num 3.27
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 32.3 13.4
#>   .. .. .. .. ..@ area   : num 3.17
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:378, 1:2] 32.5 32.5 32.5 32.5 32.5 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 32.3 13.4
#>   .. ..@ ID       : chr "14"
#>   .. ..@ area     : num 3.17
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 30.8 11.3
#>   .. .. .. .. ..@ area   : num 6.49
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:451, 1:2] 31.7 31.7 31.7 31.7 31.7 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 30.8 11.3
#>   .. ..@ ID       : chr "15"
#>   .. ..@ area     : num 6.49
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
#>   .. .. .. .. ..@ labpt  : num [1:2] 28.4 11.8
#>   .. .. .. .. ..@ area   : num 9.47
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:566, 1:2] 27.5 27.5 27.5 27.5 27.8 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 28.4 11.8
#>   .. ..@ ID       : chr "16"
#>   .. ..@ area     : num 9.47
#>  $ :Formal class 'Polygons' [package "sp"] with 5 slots
#>   .. ..@ Polygons :List of 1
#>   .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
```

```
#>   .. .. .. .. ..@ labpt  : num [1:2] 29.7 14.8
#>   .. .. .. .. ..@ area   : num 15.7
#>   .. .. .. .. ..@ hole   : logi FALSE
#>   .. .. .. .. ..@ ringDir: int 1
#>   .. .. .. .. ..@ coords : num [1:385, 1:2] 26.9 27.1 27.2 27.5 28.6 ...
#>   .. ..@ plotOrder: int 1
#>   .. ..@ labpt    : num [1:2] 29.7 14.8
#>   .. ..@ ID       : chr "17"
#>   .. ..@ area     : num 15.7
```

The first thing we learn from the output of this command is that the `sudan01@polygons` is a list with a length of 18.

In this case, we can then use our knowledge of the class list to our advantage to be able to handle `sudan01@polygons`.

Lists are convenient class or mode of data because they can accommodate multiple objects within them and these objects don't have to be of the same mode / class or length. Lists are of single dimension each of which refer to a object or dataset that has been included into the list.

This means that we can use **R**'s powerful subscripting (sometimes referred to as indexing) capabilities to access specific components of `sudan01@polygons`. We can try this by running the following command:

```
sudan01@polygons[1]
```

In this command we are instructing **R** to give us the first element / component in the list `sudan01@polygons`. This gives us the dataset and hints on the structure of the first *polygon* in the list of 18 *polygons* of `sudan01` **shapefile** (see below).

If we want to view the dataset for the 11th *polygon* in the list, we use:

```
sudan01@polygons[11]
```

If we want to view the dataset for the 6th and 7th *polygon* in the list, we use:

```
sudan01@polygons[6:7]
```

or

```
sudan01@polygons[c(6,7)]
```

If we want to view the dataset for the 3rd and the 16th *polygon* on the list, we use:

```
sudan01@polygons[c(3,16)]
```

Let us now try to extract the `plotOrder` slot of `sudan01` **shapefile**. We make a call as follows:

```
sudan01@plotOrder
```

which gives this result:

```
#>  [1] 12  3 13 18 11 17  5 16  8  4  9 14 15  7  2  6  1 10
```

The output is a numeric vector of length 18. The values in the vector represent the *polygon* that gets plotted first. In this case, the 13th *polygon* is the first to be plotted and the 1st *polygon* will be the last to be plotted.

The `plotOrder` slot is linked or related to the area slot of each of the *polygon* found in the dataset for each of the *polygons* in `sudan01` object. The *polygon* with the biggest area gets plotted first and the one with the smallest area gets plotted last.

Given this, it would be a good exercise of our newly learned skills of handling and manipulating mapping data to try to create a dataframe that combines the `sudan01@plotOrder` vector with the corresponding area size of each *polygon* referred to in the plotting order. This is a good way of checking whether indeed the plotting order of *polygons* is based on decreasing area size. We do this by:

```
areaAll <- NULL

for(i in 1:length(sudan01)) {
  area <- sudan01@polygons[sudan01@plotOrder][[i]]@Polygons[[1]]@area
  areaAll <- c(areaAll, area)
}

orderByArea <- data.frame(sudan01@plotOrder, areaAll)
```

The result of this is:

```
#>    sudan01.plotOrder   areaAll
#> 1                 12 31.366210
#> 2                  3 26.736778
#> 3                 13 18.573504
#> 4                 18 15.716028
#> 5                 11 11.146239
#> 6                 17  9.469990
#> 7                  5  6.905019
#> 8                 16  6.491091
#> 9                  8  4.985051
#> 10                 4  4.436722
#> 11                 9  4.102905
#> 12                14  0.001525
#> 13                15  3.169614
#> 14                 7  3.160318
#> 15                 2  2.745432
#> 16                 6  2.277623
#> 17                 1  1.859501
#> 18                10  1.790615
```

Let us now take a look at the `bbox` slot of `sudan01`. We make a call as follows:

```
sudan01@bbox
```

This gives the following results:

```
sudan01@bbox
```

This is basically telling us that the **shapefile** has a minimum `x` coordinate value of 21.8131148 and a maximum `x` coordinate value of 38.5909212. For the `y` coordinates, the **shapefile** has a minimum of 8.6411405 and a maximum of 23.1428919.

Another way of getting the minimum and maximum `x` and `y` coordinates of a **shapefile** is by using the `bbox()` function. It can be used as follows:

```
sudan01Limits <- bbox(sudan01)
```

The object `sudan01Limits` is equivalent to `sudan01@bbox`.

This minimum and maximum `x` and `y` coordinates is for the entire **shapefile**.

Finally, let us take a look at the `proj4string` slot of `sudan01`. This can be retrieved by calling the following:

```
sudan01@proj4string
```

The result of this command is:

```
sudan01@proj4string
```

This indicates that the projection of the **shapefile** is longitude and latitude based on **datum WGS84**.

Another way of getting the projection is by using the `proj4string()` function as follows:

```
proj4string(sudan01)
```

This gives the same result but with a slightly different format.

```
#> [1] "+proj=utm +zone=36 +a=6378249.2 +b=6356514.999904194 +units=m +no_defs"
```

2 Plotting maps

3 Manipulating shapefile data