Mapping with R

Ernest Guevarra

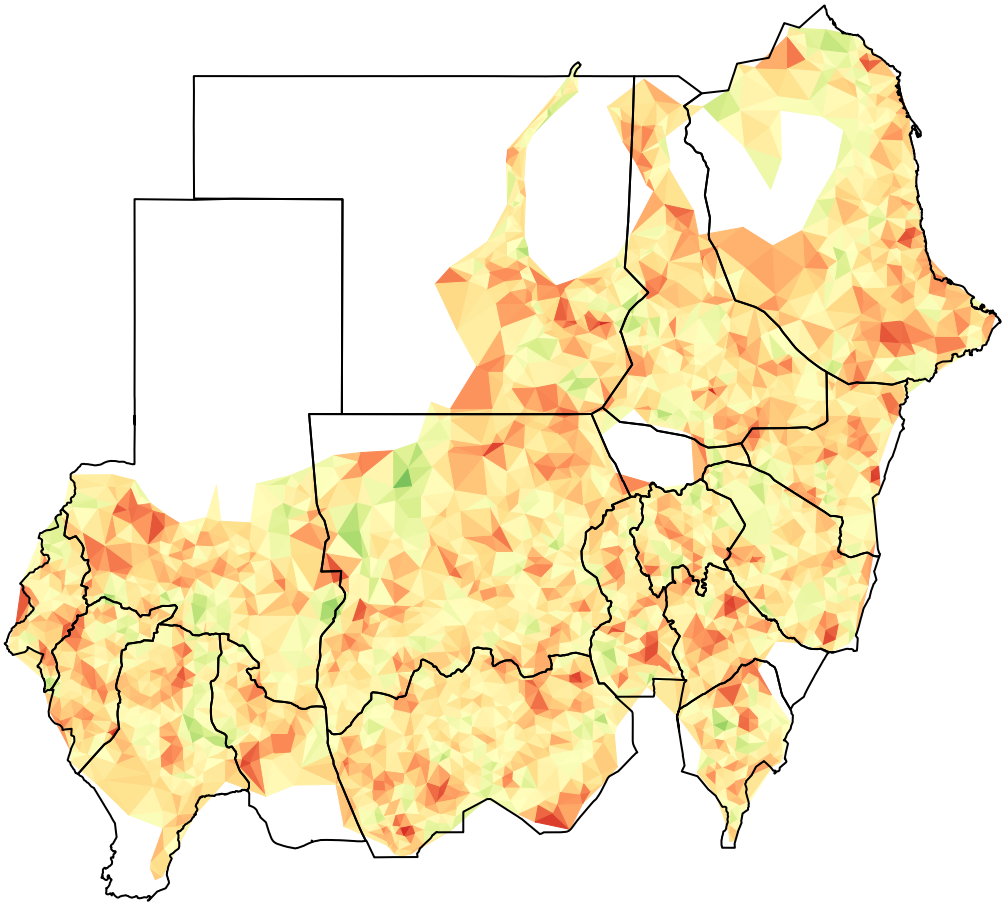
2018-10-09

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=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=
```

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 STATEAR
#> 0        West Darfur       West Darfur       West Darfur    <NA>    <NA>
#> 1     Central Darfur       West Darfur    Central Darfur    <NA>    <NA>
#> 2       North Darfur      North Darfur      North Darfur    <NA>    <NA>
#> 3       East Darfur      South Darfur       East Darfur    <NA>    <NA>
#> 4       South Darfur      South Darfur      South Darfur    <NA>    <NA>
#> 5          Al Gezira              <NA>              <NA>    <NA>    <NA>
#> 6          Blue Nile              <NA>              <NA>    <NA>    <NA>
#> 7            Gedaref              <NA>              <NA>    <NA>    <NA>
#> 8            Kassala              <NA>              <NA>    <NA>    <NA>
#> 9           Khartoum              <NA>              <NA>    <NA>    <NA>
#> 10              Nile              <NA>              <NA>    <NA>    <NA>
```

```
#> 11          Northern              <NA>              <NA>    <NA>    <NA>
#> 12           Red Sea              <NA>              <NA>    <NA>    <NA>
#> 13            Sennar              <NA>              <NA>    <NA>    <NA>
#> 14        White Nile              <NA>              <NA>    <NA>    <NA>
#> 15 Southern Kordofan Southern Kordofan Southern Kordofan    <NA>    <NA>
#> 16  Western Kordofan    South Kordofan  Western Kordofan    <NA>    <NA>
#> 17 Northern Kordofan Northern Kordofan Northern Kordofan    <NA>    <NA>
#>    Source STATE_CODE ISO_CODE
#> 0    <NA>       <NA>     <NA>
#> 1    <NA>       <NA>     <NA>
#> 2    <NA>       <NA>     <NA>
#> 3    <NA>       <NA>     <NA>
#> 4    <NA>       <NA>     <NA>
#> 5     CBS       SU04    SD-07
#> 6     CBS       SU02    SD-24
#> 7     CBS       SU05    SD-06
#> 8     CBS       SU07    SD-05
#> 9     CBS       SU08    SD-03
#> 10    CBS       SU10    SD-04
#> 11    CBS       SU11    SD-01
#> 12    CBS       SU15    SD-26
#> 13    CBS       SU16    SD-25
#> 14    CBS       SU25    SD-08
#> 15   <NA>       <NA>     <NA>
#> 16   <NA>       <NA>     <NA>
#> 17   <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.

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:

```
#>        min      max
#> x 21.81311 38.59092
#> y  8.64114 23.14289
```

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=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"
```

2 Plotting maps

In this exercise, we will learn how to plot maps using your existing knowledge of basic plotting functions and new lessons on additional plotting functions specific to maps.

By now you would be familiar already with the `plot()` function which is the most basic of plotting functions. For maps, we use `plot()` in the same way as we would for other datasets. The main difference is that the methods used in the plotting functions for **shapefiles** are set by the `sp` package and is optimised for mapping purposes.

Therefore, plotting maps is as easy as calling the `plot()` function and passing the `spatial` object to it as shown below.

```
plot(sudan01)
```

This produces the following map:

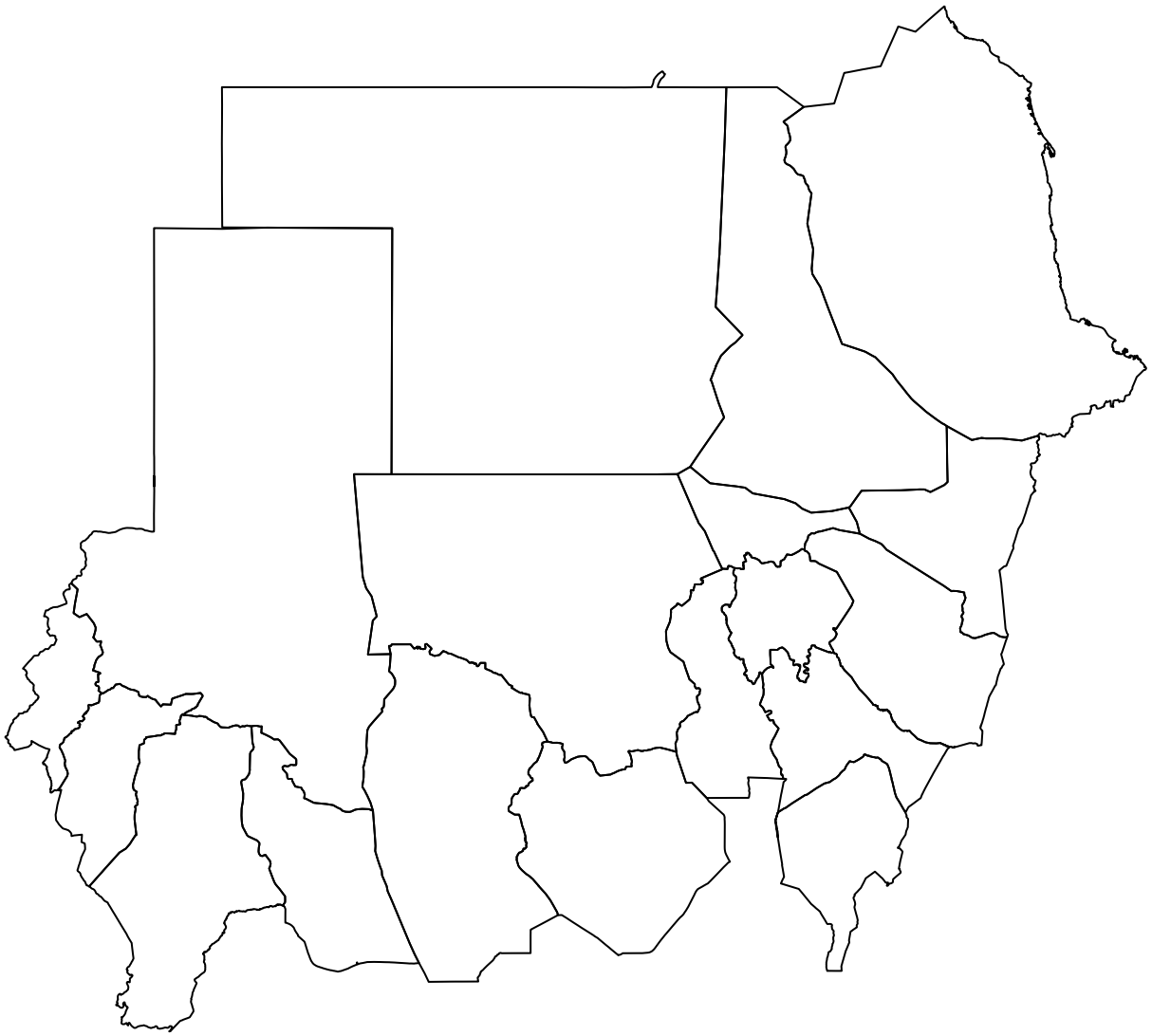


Figure 2.1: Map of the states of Sudan with default plotting options

From this plot of `sudan01` `SpatialPolygonsDataFrame` object, we learn about some of the default plotting parameters under the `sp` package. *Line type* (`lty`) is set as a solid line (`lty = 1`) and *line width* (`lwd`) is set at `lwd = 1`. *Border colour* (`border`) is set as `border = black` while the *fill colour* (`col`) is set at `col = NA`. We should remember that the class of `sudan01` object is a `SpatialPolygonsDataFrame` hence it is composed of closed rings that are not holes (i.e. empty). Hence, in the plotting parameters, border and fill colours are specified separately.

Let us now try to plot another map of `sudan01` object but this time change some of the default plotting parameters.

```
plot(sudan01, lty = 1, lwd = 2, border = "blue", col = "gray90")
```

This command gives us the following map of Sudan:

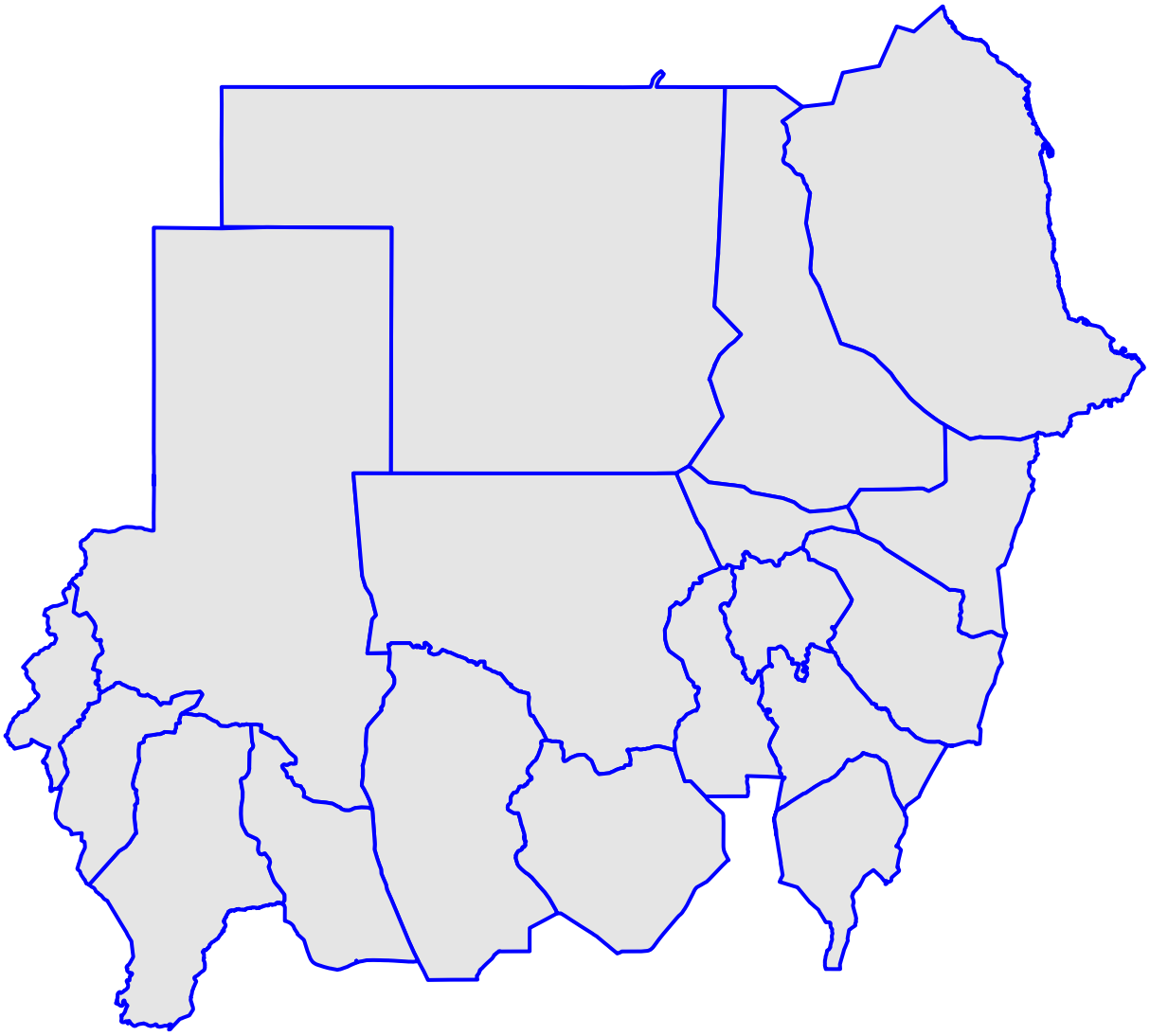


Figure 2.2: Map of the states of Sudan with new plotting parameters

The map above illustrates the changes in the plot created by the changes in the plotting parameters that we have specified. The width of the border is now thicker and its colour is now blue. The inside of the map is now coloured gray.

So far we have focused a lot on `sudan01` object. Hence we have learned a lot about the features, structure and characteristics of a **polygon** and a `SpatialPolygonsDataFrame` that contains **polygons** of each of the states of Sudan. However, we have yet to work with another

`SpatialPolygonsDataFrame` object named `sudan02` which is a collection of **polygons** for each of the localities in Sudan.

What we will try to learn now is how to work with the `sudan01` and `sudan02` objects to create a map of Sudan that shows the different localities and also shows how these localities are grouped in order to make up each of the states in Sudan.

To be able to complete this task, we need to learn about *map layers*.

The idea of *map layers* is that different set of map features (i.e. roads, topography, watery systems) that usually have separate **shapefile** datasets are put together in a single map by plotting each of these features on top of each other in layers. Depending on the type and class of the **shapefile** objects being layered, the order by which a feature is layered on to another determines which features of which layer is visible or given emphasis.

Let us now try to layer `sudan01` and `sudan02` objects such that the base map is that of the localities and on top is that of the state boundaries. The final map will then show divisions by state and then further subdivisions per state by localities. We call the following commands to produce the map below:

```
#
# Plot localities of Sudan
#
plot(sudan02, border = "gray")
#
# Plot states of Sudan
#
plot(sudan01, lwd = 2, add = TRUE)
```

This produces the following map.

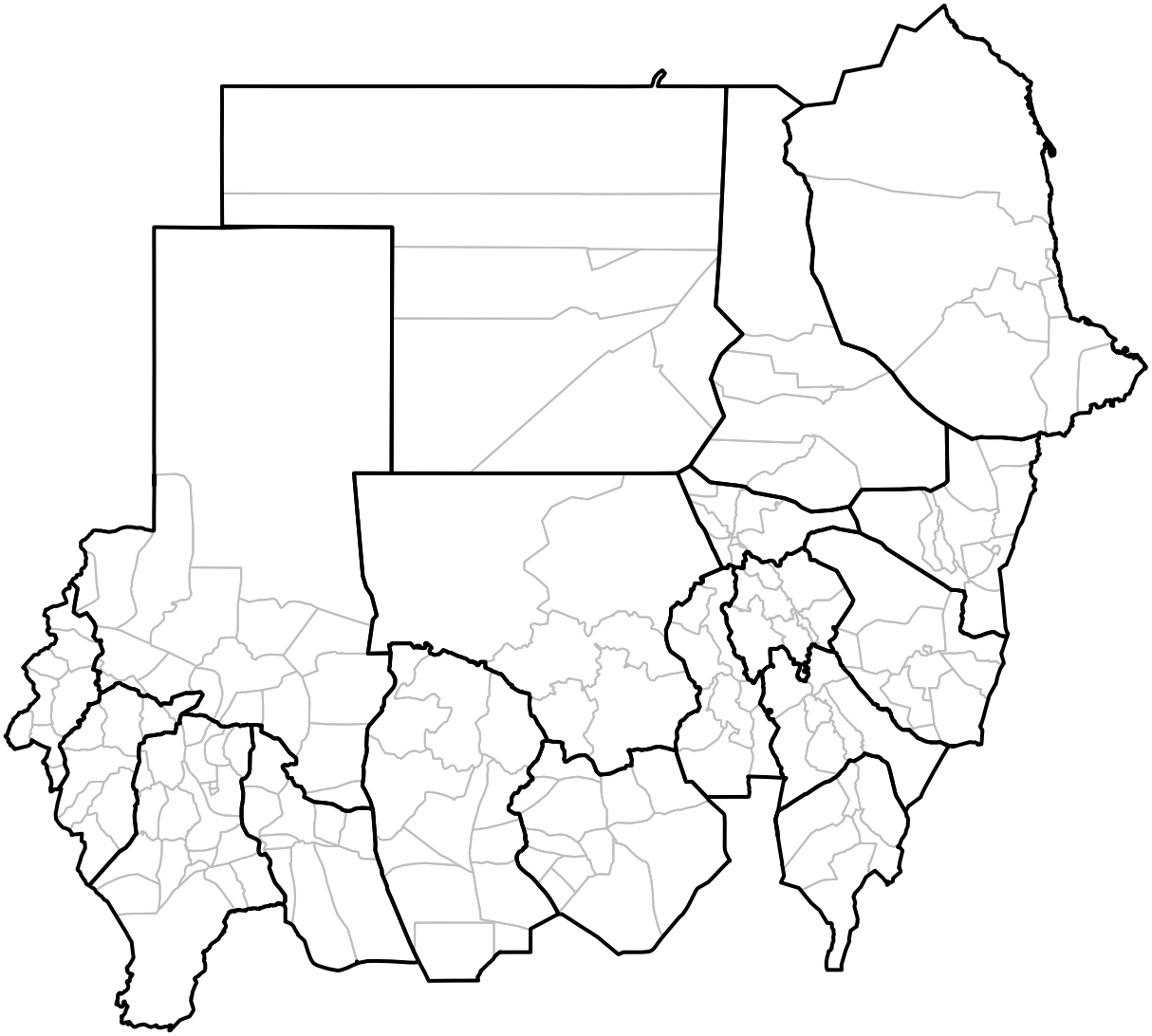


Figure 2.3: Map of localities and states of Sudan

Now, let us learn how to add layers of other **shapefile** types/classes onto the base Sudan maps.

Earlier in Exercise 1, we have retrieved the dataset for the grid used in the Sudan S3M I and assigned it to the object `grid12poly` and `grid12kmSudan`. The difference between these two objects is that the first one is a `SpatialPolygonsDataFrame` while the second is a `SpatialLinesDataFrame`.

We will focus our attention to `grid12kmSudan` to familiarise ourselves with the `SpatialLinesDataFrame` class objects. Let us look at the structure of a `SpatialLinesDataFrame`.

```
str(grid12kmSudan)
```

The results of this command is as expected quite long. But looking at the first line of the output, we learn that `grid12kmSudan` has 4 slots.

```
#> Formal class 'SpatialLinesDataFrame' [package "sp"] with 4 slots
#>   ..@ data       :'data.frame':  245 obs. of  2 variables:
#>   .. ..$ ID   : chr [1:245] "0" "1" "2" "3" ...
#>   .. ..$ COORD: num [1:245] 23.1 23 23 22.9 22.8 ...
#>   ..@ lines      :List of 245
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 23.1 23.1
#>   .. .. .. ..@ ID   : chr "0"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 23 23
#>   .. .. .. ..@ ID   : chr "1"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 23 23
#>   .. .. .. ..@ ID   : chr "2"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 22.9 22.9
#>   .. .. .. ..@ ID   : chr "3"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 22.8 22.8
#>   .. .. .. ..@ ID   : chr "4"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 22.7 22.7
#>   .. .. .. ..@ ID   : chr "5"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 22.6 22.6
#>   .. .. .. ..@ ID   : chr "6"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 22.5 22.5
#>   .. .. .. ..@ ID   : chr "7"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 22.4 22.4
#>   .. .. .. ..@ ID   : chr "8"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 22.3 22.3
#>   .. .. .. ..@ ID   : chr "9"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 22.2 22.2
#>   .. .. .. ..@ ID   : chr "10"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 22.1 22.1
#>   .. .. .. ..@ ID   : chr "11"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 22 22
#>   .. .. .. ..@ ID   : chr "12"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21.9 21.9
#>   .. .. .. ..@ ID   : chr "13"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21.8 21.8
#>   .. .. .. ..@ ID   : chr "14"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21.7 21.7
#>   .. .. .. ..@ ID   : chr "15"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21.6 21.6
#>   .. .. .. ..@ ID   : chr "16"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21.5 21.5
#>   .. .. .. ..@ ID   : chr "17"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21.4 21.4
#>   .. .. .. ..@ ID   : chr "18"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21.4 21.4
#>   .. .. .. ..@ ID   : chr "19"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21.3 21.3
#>   .. .. .. ..@ ID   : chr "20"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21.2 21.2
#>   .. .. .. ..@ ID   : chr "21"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21.1 21.1
#>   .. .. .. ..@ ID   : chr "22"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 21 21
#>   .. .. .. ..@ ID   : chr "23"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 20.9 20.9
#>   .. .. .. ..@ ID   : chr "24"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 20.8 20.8
#>   .. .. .. ..@ ID   : chr "25"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 20.7 20.7
#>   .. .. .. ..@ ID   : chr "26"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 20.6 20.6
#>   .. .. .. ..@ ID   : chr "27"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 20.5 20.5
#>   .. .. .. ..@ ID   : chr "28"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 20.4 20.4
#>   .. .. .. ..@ ID   : chr "29"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 20.3 20.3
#>   .. .. .. ..@ ID   : chr "30"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 20.2 20.2
#>   .. .. .. ..@ ID   : chr "31"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 20.1 20.1
#>   .. .. .. ..@ ID   : chr "32"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 20 20
#>   .. .. .. ..@ ID   : chr "33"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19.9 19.9
#>   .. .. .. ..@ ID   : chr "34"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19.9 19.9
#>   .. .. .. ..@ ID   : chr "35"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19.8 19.8
#>   .. .. .. ..@ ID   : chr "36"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19.7 19.7
#>   .. .. .. ..@ ID   : chr "37"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19.6 19.6
#>   .. .. .. ..@ ID   : chr "38"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19.5 19.5
#>   .. .. .. ..@ ID   : chr "39"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19.4 19.4
#>   .. .. .. ..@ ID   : chr "40"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19.3 19.3
#>   .. .. .. ..@ ID   : chr "41"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19.2 19.2
#>   .. .. .. ..@ ID   : chr "42"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19.1 19.1
#>   .. .. .. ..@ ID   : chr "43"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 19 19
#>   .. .. .. ..@ ID   : chr "44"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18.9 18.9
#>   .. .. .. ..@ ID   : chr "45"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18.8 18.8
#>   .. .. .. ..@ ID   : chr "46"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18.7 18.7
#>   .. .. .. ..@ ID   : chr "47"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18.6 18.6
#>   .. .. .. ..@ ID   : chr "48"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18.5 18.5
#>   .. .. .. ..@ ID   : chr "49"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18.4 18.4
#>   .. .. .. ..@ ID   : chr "50"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18.3 18.3
#>   .. .. .. ..@ ID   : chr "51"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18.3 18.3
#>   .. .. .. ..@ ID   : chr "52"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18.2 18.2
#>   .. .. .. ..@ ID   : chr "53"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18.1 18.1
#>   .. .. .. ..@ ID   : chr "54"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 18 18
#>   .. .. .. ..@ ID   : chr "55"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 17.9 17.9
#>   .. .. .. ..@ ID   : chr "56"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 17.8 17.8
#>   .. .. .. ..@ ID   : chr "57"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 17.7 17.7
#>   .. .. .. ..@ ID   : chr "58"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 17.6 17.6
#>   .. .. .. ..@ ID   : chr "59"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 17.5 17.5
#>   .. .. .. ..@ ID   : chr "60"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 17.4 17.4
#>   .. .. .. ..@ ID   : chr "61"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 17.3 17.3
#>   .. .. .. ..@ ID   : chr "62"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 17.2 17.2
#>   .. .. .. ..@ ID   : chr "63"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 17.1 17.1
#>   .. .. .. ..@ ID   : chr "64"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 17 17
#>   .. .. .. ..@ ID   : chr "65"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16.9 16.9
#>   .. .. .. ..@ ID   : chr "66"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16.8 16.8
#>   .. .. .. ..@ ID   : chr "67"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16.7 16.7
#>   .. .. .. ..@ ID   : chr "68"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16.7 16.7
#>   .. .. .. ..@ ID   : chr "69"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16.6 16.6
#>   .. .. .. ..@ ID   : chr "70"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16.5 16.5
#>   .. .. .. ..@ ID   : chr "71"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16.4 16.4
#>   .. .. .. ..@ ID   : chr "72"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16.3 16.3
#>   .. .. .. ..@ ID   : chr "73"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16.2 16.2
#>   .. .. .. ..@ ID   : chr "74"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16.1 16.1
#>   .. .. .. ..@ ID   : chr "75"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 16 16
#>   .. .. .. ..@ ID   : chr "76"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15.9 15.9
#>   .. .. .. ..@ ID   : chr "77"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15.8 15.8
#>   .. .. .. ..@ ID   : chr "78"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15.7 15.7
#>   .. .. .. ..@ ID   : chr "79"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15.6 15.6
#>   .. .. .. ..@ ID   : chr "80"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15.5 15.5
#>   .. .. .. ..@ ID   : chr "81"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15.4 15.4
#>   .. .. .. ..@ ID   : chr "82"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15.3 15.3
#>   .. .. .. ..@ ID   : chr "83"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15.2 15.2
#>   .. .. .. ..@ ID   : chr "84"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15.1 15.1
#>   .. .. .. ..@ ID   : chr "85"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15.1 15.1
#>   .. .. .. ..@ ID   : chr "86"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 15 15
#>   .. .. .. ..@ ID   : chr "87"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 14.9 14.9
#>   .. .. .. ..@ ID   : chr "88"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 14.8 14.8
#>   .. .. .. ..@ ID   : chr "89"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 14.7 14.7
#>   .. .. .. ..@ ID   : chr "90"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 14.6 14.6
#>   .. .. .. ..@ ID   : chr "91"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 14.5 14.5
#>   .. .. .. ..@ ID   : chr "92"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
```

```
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 14.4 14.4
#>   .. .. .. ..@ ID   : chr "93"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 14.3 14.3
#>   .. .. .. ..@ ID   : chr "94"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 14.2 14.2
#>   .. .. .. ..@ ID   : chr "95"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 14.1 14.1
#>   .. .. .. ..@ ID   : chr "96"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 14 14
#>   .. .. .. ..@ ID   : chr "97"
#>   .. ..$ :Formal class 'Lines' [package "sp"] with 2 slots
#>   .. .. .. ..@ Lines:List of 1
#>   .. .. .. .. ..$ :Formal class 'Line' [package "sp"] with 1 slot
#>   .. .. .. .. .. .. ..@ coords: num [1:2, 1:2] 21.8 38.6 13.9 13.9
#>   .. .. .. ..@ ID   : chr "98"
#>   .. .. [list output truncated]
#>   ..@ 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=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=
```

The 4 slots in a `SpatialLinesDataFrame` are:

Table 2.1: SpatialLinesDataFrame structure

data	Contains the index or reference data frame of the **shapefile** the number of rows of which indicates the number of *lines* that comprise the entire **shapefile**
lines	Contains n number of datasets based on the number of *lines* that comprise the entire **shapefile**.
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

Compared to the `SpatialPolygonsDataFrame`, `grid12kmSudan` has a slot for *lines* (as expected) but doesn't have the `plotOrder` slot. If we look back at what we've learned regarding `plotOrder`, it is understandable why this is not included in a `SpatialLinesDataFrame` object. The `plotOrder` is determined by the area of the shapes in the object. However, `SpatialLinesDataFrame` objects does not have a value for area because it is just composed of *lines*, not *polygons*.

A closer look at the slot `grid12kmSudan@lines`, we notice that the data structure is much simpler than that of *polygons*. Each *line* is simply defined by a set of 2 points.

Let us now add the `grid12kmSudan` object as another *layer* on our map. We should put it on top of `sudan02` and `sudan01` with a thin *line width* (`< 1`) and a light colour. Here is the command that will produce this.

```
plot(sudan02, border = "gray")
plot(sudan01, lwd = 2, add = TRUE)
plot(grid12kmSudan, lwd = 0.5, col = "gray90", add = TRUE)
```

This command produces the following map:

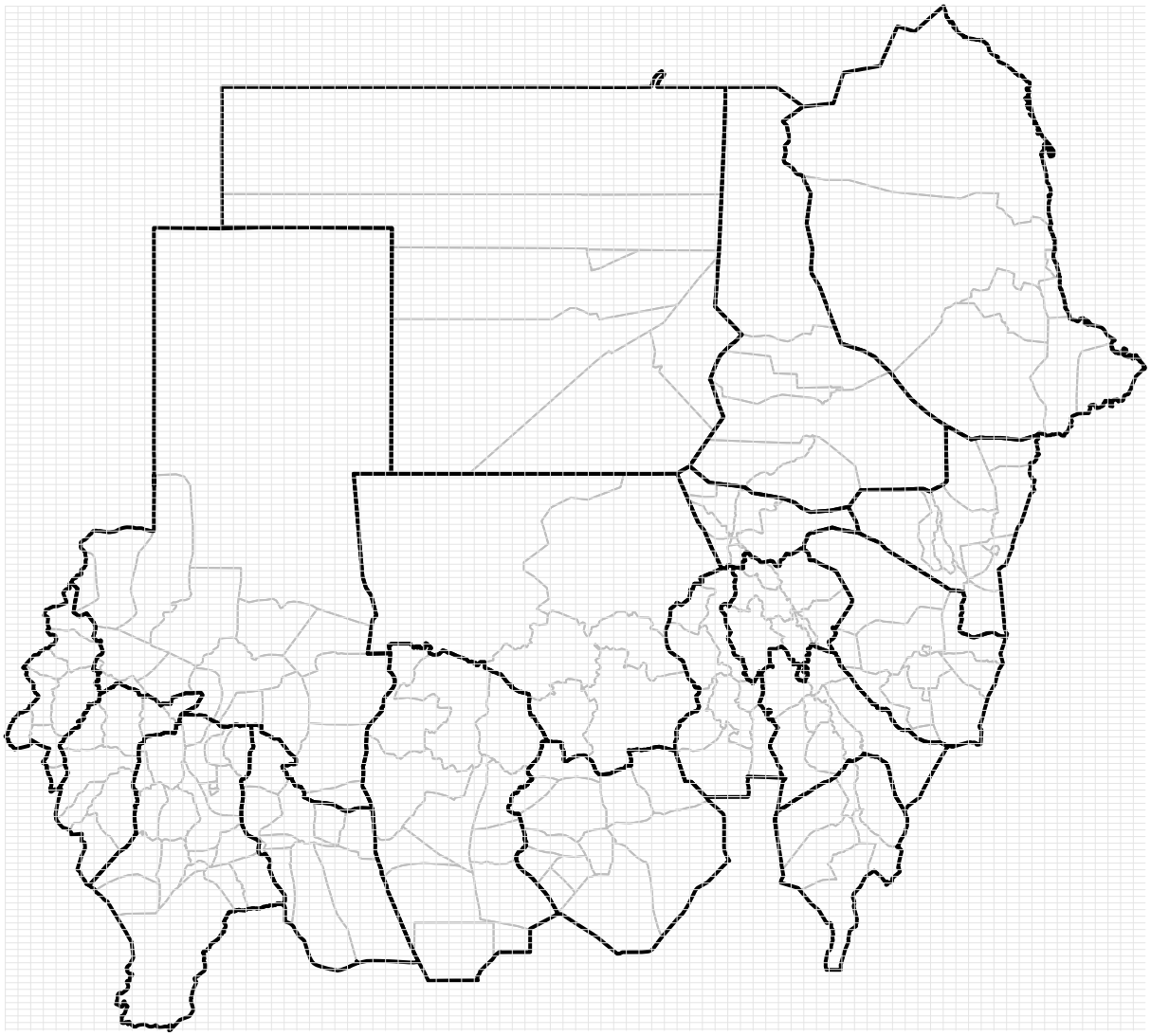


Figure 2.4: Map of localities and states of Sudan with rectangular grid

Now let us learn about points data.

Earlier, we have created an object called `villages`. This object contains village data for the whole of Sudan with their `x` and `y` geographical coordinates.

If we check the class of the object `villages`

```
class(villages)
```

we find out that this object is a data.frame and not an sp class object.

```
#> [1] "data.frame"
```

However, this object is still map data because it contains geographic information which in this case is the `x` and `y` coordinates of the villages of Sudan.

This is the only information we need to be able to plot points data on the map. We can do this by calling the following commands:

```
plot(sudan02, border = "gray")
plot(sudan01, lwd = 2, add = TRUE)
plot(grid12kmSudan, lwd = 0.5, col = "gray90", add = TRUE)
points(villages$X, villages$Y, pch = 20, cex = 0.2, col = "gray90")
```

This produces the following map.

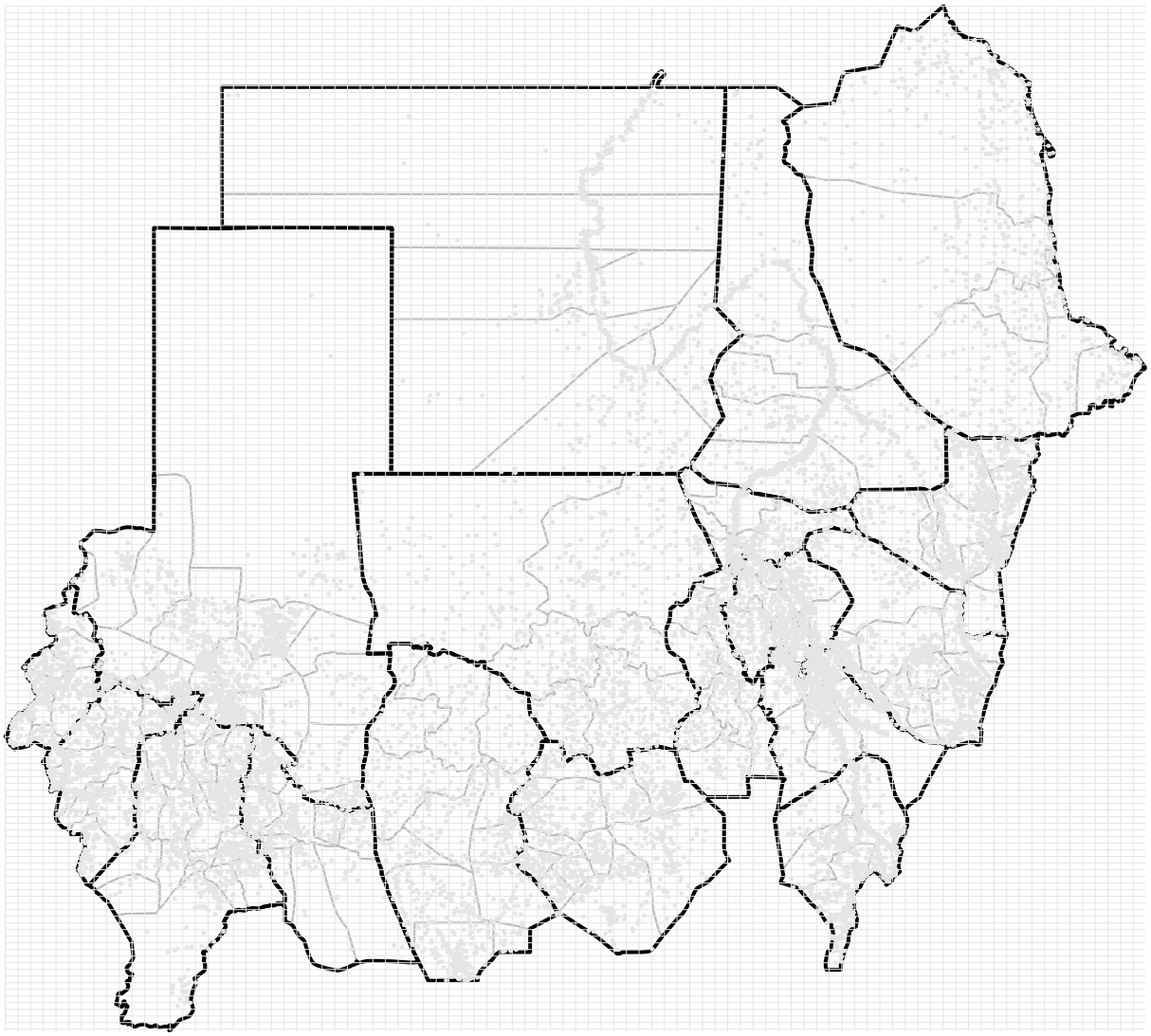


Figure 2.5: Map of localities and states of Sudan with rectangular grid and villages

In the commands above, we learn a new function called `points()` and some new graphical parameters that are useful in plotting *points*.

The `points()` function basically instructs **R** to plot a series of *points* given the specified coordinates. The kind of character that is drawn to represent the *point* is determined by the parameter called `pch` (*point character*) with default being a hollow circle.

The most commonly used point characters are specified by the following numeric values:

In this case, we used `pch = 20` which is a solid point that is 2/3 the size of `pch = 19`.

The other graphical parameter that we learn here is `cex` (*character expansion*) which determines the size of the point character and has a default value of `cex = 1`.

By now you would have noticed that when we used the `points()` function, we didn't have to add an argument to add the points on to the current graphical device. By default, the `points()` function will draw the points on to the current graphical device. In fact, `points()` will give an error message and will not plot anything if there is no open graphical device as you see below.

```
points(villages$X, villages$Y, pch = 20, cex = 0.2, col = "gray90")
```

Let us now learn how to add other map components onto the current map. These may include

Table 2.2: Other layers that can be added to maps

labels	Either a numeric or character identifier that names the different features on the map
legend	Guide to what the shapes and symbols on the map refer to or mean
scale	The scale of a map is the ratio of a distance on the map to the corresponding distance on the ground.
compass	Indicates where the north, east, west and south directions are on the map

First, let us try adding labels to our map.

For `sudan01` object which contains a map of Sudan divided into states, we would like to be able to label each state accordingly. We would like this label to be placed at the centre of each state polygon as this is based on standard mapping conventions. To do this, we will use two functions that we have not learned yet. These are `coordinates()` and `text()`.

The `coordinates()` function is a function made available in the `sp` package. This function returns the centroid of the **polygon** or **polygons** in a map object. The `coordinates()` function returns a matrix containing the `x` and `y` coordinates of the centroid of the **polygons** in the map object. If there is only 1 **polygon** in the map object, then the resulting matrix is a single row.

The `text()` function on the other hand adds text into the current plotting device based on a specific `x` and `y` coordinates which for the purpose of labelling will be the centroid provided by the `coordinates()` function.

Additional parameters required for `text()` are the labels which is a character vector specifying the text to be written. In this case, we need the names of the states. We can easily get this from the data slot in `sudan01`. We use the parameter `pos` to specify where and how far the text will be written in relation to the `x` and `y` coordinates specified. The `pos` parameter can take values of `1`, `2`, `3`, or `4` which places the text at the *bottom*, *left*, *top* or *right* of the centroid respectively. The `offset` parameter determines how far from the centroid the text will be written. The `cex` parameter sets the size of the text to be written.

We apply this function to label the states by giving the following commands:

```
sudan01Centroids <- coordinates(sudan01)
plot(sudan01)
text(x = sudan01Centroids[ , 1],
     y = sudan01Centroids[ , 2],
     labels = sudan01@data$State,
     pos = 3,
     offset = 0.2,
     cex = 0.8)
```

This results in the following map:

Figure 2.6: Map of states of Sudan with labels

Now we shall add a *legend* on the map.

We use the `legend()` function for this. Following are the commands to create a *legend* for the map of `sudan01`, `sudan02` and `villages`.

```
plot(sudan02, border = "gray")
plot(sudan01, lwd = 2, add = TRUE)
points(villages$X,
       villages$Y,
       pch = 20,
       cex = 0.2,
       col = "gray90")
legend("topleft",
       y.intersp = 1.2,
       legend = c("state borders", "locality borders", "villages"),
       lty = c(1, 1, NA),
       lwd = c(2, 1, NA),
       pch = c(NA, NA, 20),
       col = c("black", "gray", "gray90"),
       cex = 0.65,
       pt.cex = 1,
       bty = "o",
       bg = "white")
```

This produces the following map:

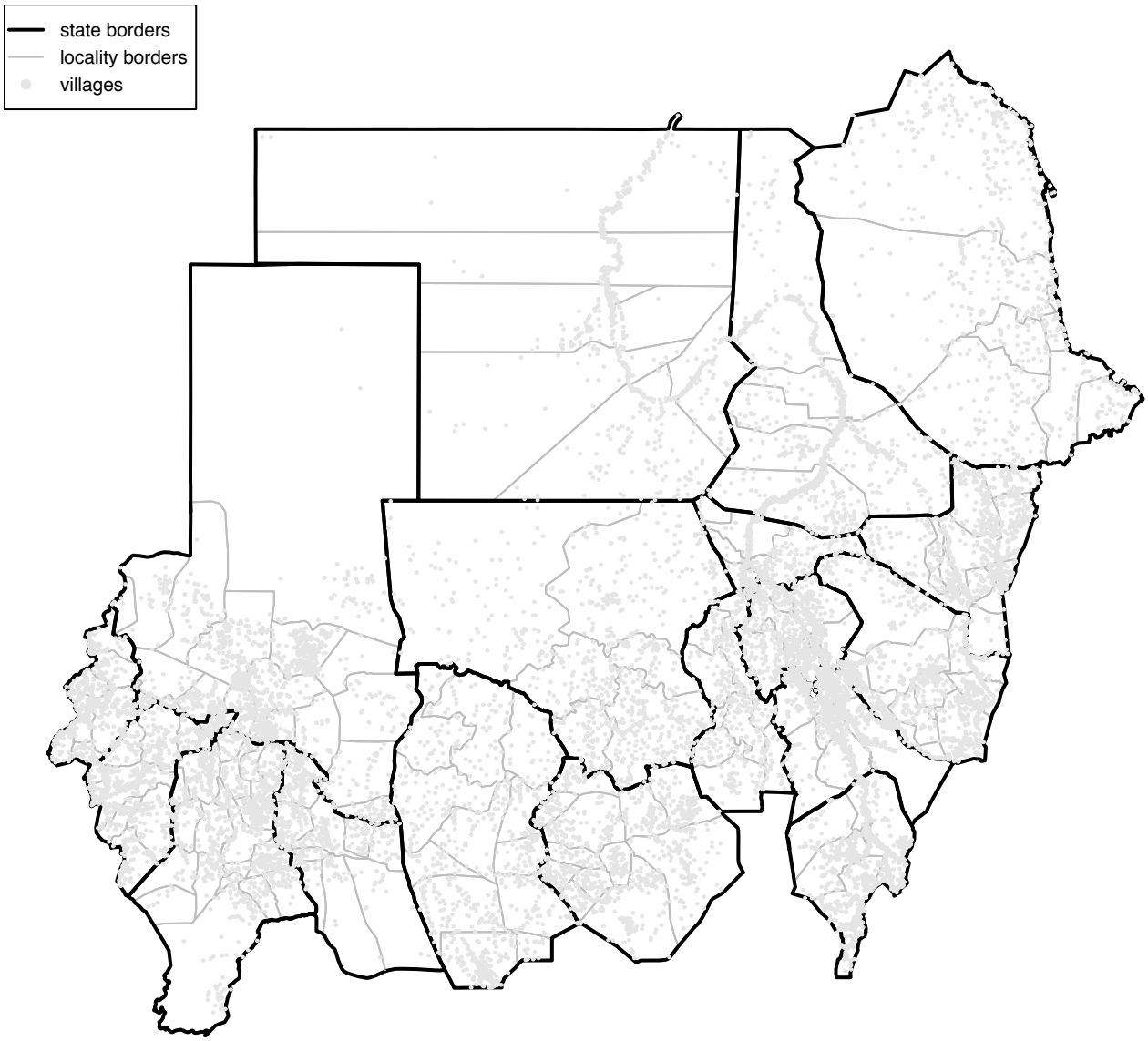


Figure 2.7: Map of localities, states and villages of Sudan with labels and legend

Lastly, we add a *scale* to our map.

We use the `map.scale()` function for this. We need to install and load the library `maps` to use this function. We add the scale on the map through the following commands:

```
plot(sudan02, border = "gray")
plot(sudan01, lwd = 2, add = TRUE)
points(villages$X, villages$Y, pch = 20, cex = 0.2, col = "gray90")
legend("topleft",
       y.intersp = 1.2,
       legend = c("state borders", "locality borders", "villages"),
       lty = c(1, 1, NA),
       lwd = c(2, 1, NA),
       pch = c(NA, NA, 20),
       col = c("black", "gray", "gray90"),
       cex = 0.65,
       pt.cex = 1,
       bty = "o",
       bg = "white")
map.scale(x = bbox(sudan01)[1,],
          y = bbox(sudan01)[2,],
          ratio = TRUE,
          cex = 0.65)
```

This produces the following map:

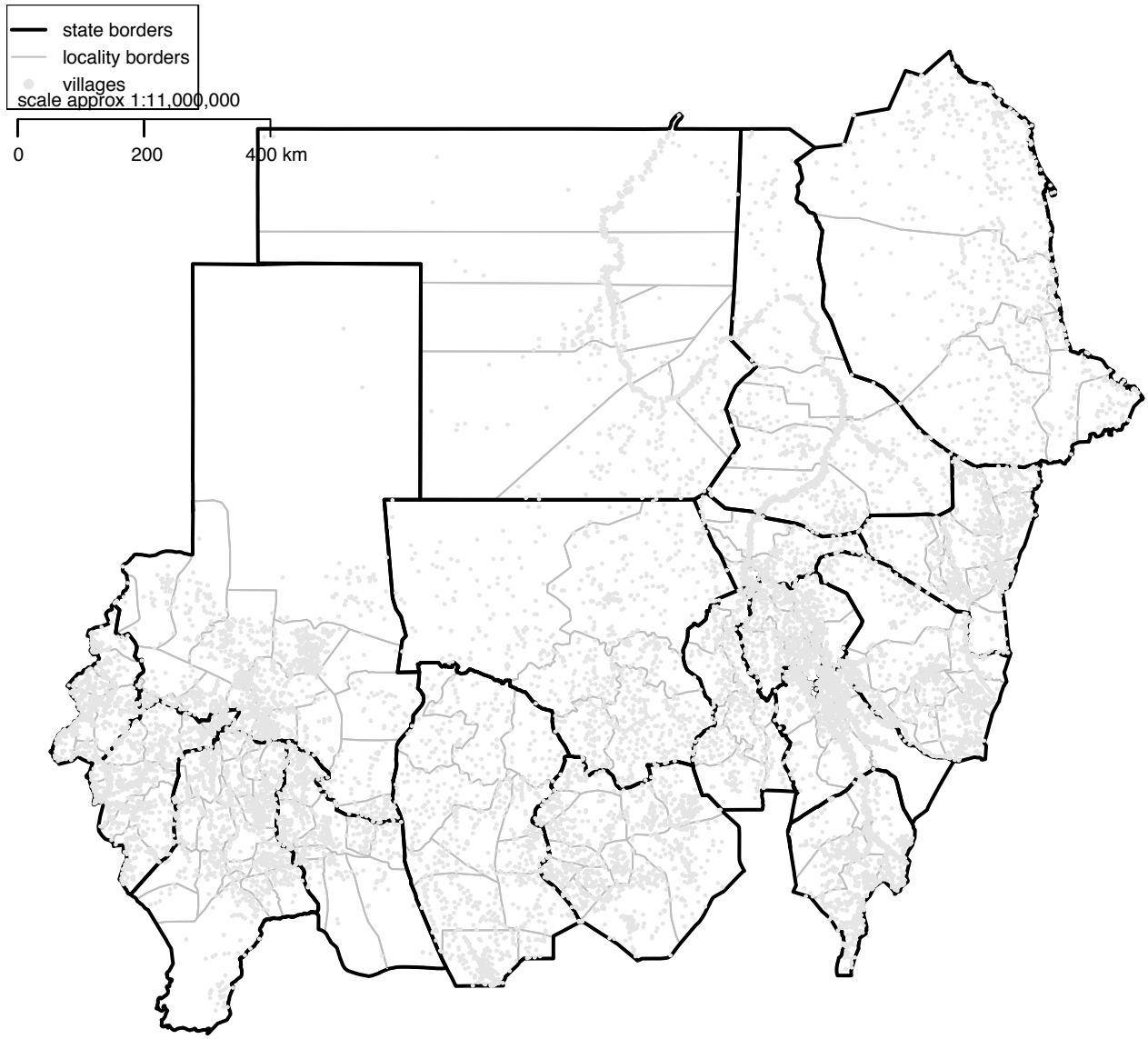


Figure 2.8: Map of localities, states and villages of Sudan with labels, legend and scale

3 Manipulating shapefile data

In this exercise, we will use what we have learned about the structure and features of shapefile data in manipulating the data and creating new map objects.

By now you would have noticed that sudan01 and sudan02 shapefile objects contain data for the whole of Sudan. This is useful for creating maps for the whole country. However, there are times when we would like to create maps of the specific states only or assign colours for certain states differently rather than a single colour for all states.

To do these maps does not require new shapefiles for each of the state. All this needs is some manipulation of the full country map dataset. This is possible because the data structure of shapefiles contains all the data required to map all the features of the overall map separately.

First, let us try to create a map of sudan01 and sudan02 with the states and the localities coloured differently from each other.

A colour palette should be created which would provide colours for each of the states. This can be done using the following commands:

```
createPalette <- colorRampPalette(colors = c("#FBB4AE", "#B3CDE3", "#CCEBC5",
                                              "#DECBE4", "#FED9A6", "#FFFFCC",
                                              "#E5D8BD", "#FDDAEC", "#F2F2F2"),
                                  space = "Lab")
sudanCol <- createPalette(18)
```

What this command uses the `colorRampPalette()` function to create a colour palette of 18 unique colours (one for each state). These colours were generated from a base set of 9 colours which in turn was chosen using a qualitative colour scheme from ColorBrewer 2.0 (see http://www.colorbrewer2.org).

Once we have a colour palette to use, we now plot the map and use the colour palette to specify the parameter col. This is done as follows:

```
plot(sudan01, border = sudanCol, col = sudanCol)
plot(sudan02, lwd = 0.5, lty = 3, border = "gray", add = TRUE)
plot(sudan01, lwd = 1, border = "gray", add = TRUE)
text(x = coordinates(sudan01)[,1],
     y = coordinates(sudan01)[,2],
     labels = sudan01@data$State,
     pos = 3,
     offset = 0.2,
     cex = 0.8)
```

The resulting map is:

Figure 3.1: Map of localities and states of Sudan coloured

Now, let us try to specify a colour to just a handful of the states while the others remain without colour. This is particularly useful when you just want to highlight specific states in the map for presentation purpose or for emphasis.

For this map, we will need to use the sudan01@data slot to steer our selection of the states of interest which we can then specify a colour for. For example, if we want to show the full Sudan map and Gedaref, Northern and Western Kordofan coloured light green, we call the following commands:

```
plot(sudan01,
     lty = 0,
     col = ifelse(sudan01@data$State %in% c("Gedaref",
                                            "Northern",
                                            "Western Kordofan"),
                  "lightgreen", NA))
plot(sudan02, lwd = 0.5, lty = 3, border = "gray", add = TRUE)
plot(sudan01, lwd = 1, add = TRUE)
text(x = coordinates(sudan01)[,1],
     y = coordinates(sudan01)[,2],
     labels = sudan01@data$State,
     pos = 3,
     offset = 0.2,
     cex = 0.8)
```

The resulting map is:

Figure 3.2: Map of specific states of Sudan coloured

There are practical uses for knowing how to manipulate map data so as to change the colour of the components polygons either based on a qualitative criteria (such as different colours for each state to distinguish them from each other) or on numerical data (such as in creating a choropleth map).

The following example of simulated CMAM coverage results per state illustrates this usefullness.

We first simulate the data as follows:

```
states   <- as.vector(sudan01@data$State)
coverage <- c(95, 56, 73, 55, 19, 97,
              60, 2, 46, 31, 70, 23,
              71, 66, 38, 78, 74, 51)

exData <- data.frame(states, coverage)
names(exData) <- c("states", "coverage")
```

The exData object gives us the simulated of CMAM coverage by state. Then we specify a colour palette as follows:

```
createMapPalette <- colorRampPalette(colors = c("white", "lightgreen",
                                                "green", "darkgreen"),
                                     space = "Lab")
mapPalette <- createMapPalette(101)
```

We now can map this data using the folowing commands:

```
plot(sudan01, lty = 0, col = mapPalette[exData$coverage + 1])
plot(sudan02, lwd = 0.5, lty = 3, border = "gray", add = TRUE)
plot(sudan01, lwd = 1, add = TRUE)
text(x = coordinates(sudan01)[,1],
     y = coordinates(sudan01)[,2],
     labels = sudan01@data$State,
     pos = 3,
     offset = 0.2,
     cex = 0.8)
```

This results in the following map:

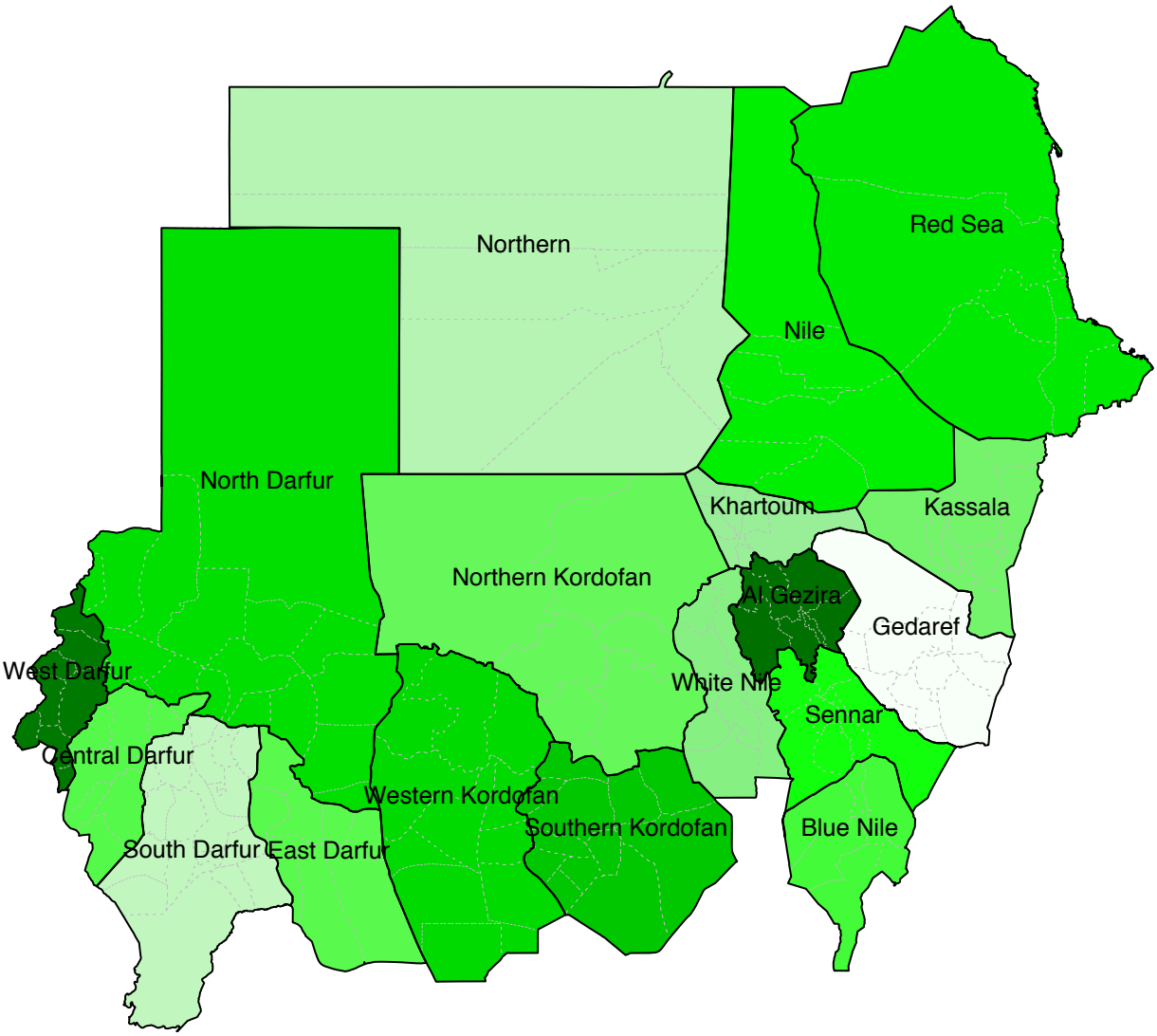


Figure 3.3: Sample coverage map

Finally, we learn how to create maps for each state separately using the `sudan01` map dataset.

To begin with, let's try to create and map a map dataset for North Darfur. We call the following commands:

```
nDarfur <- subset(sudan01, sudan01@data$State == "North Darfur")
plot(nDarfur)
```

To produce the following:

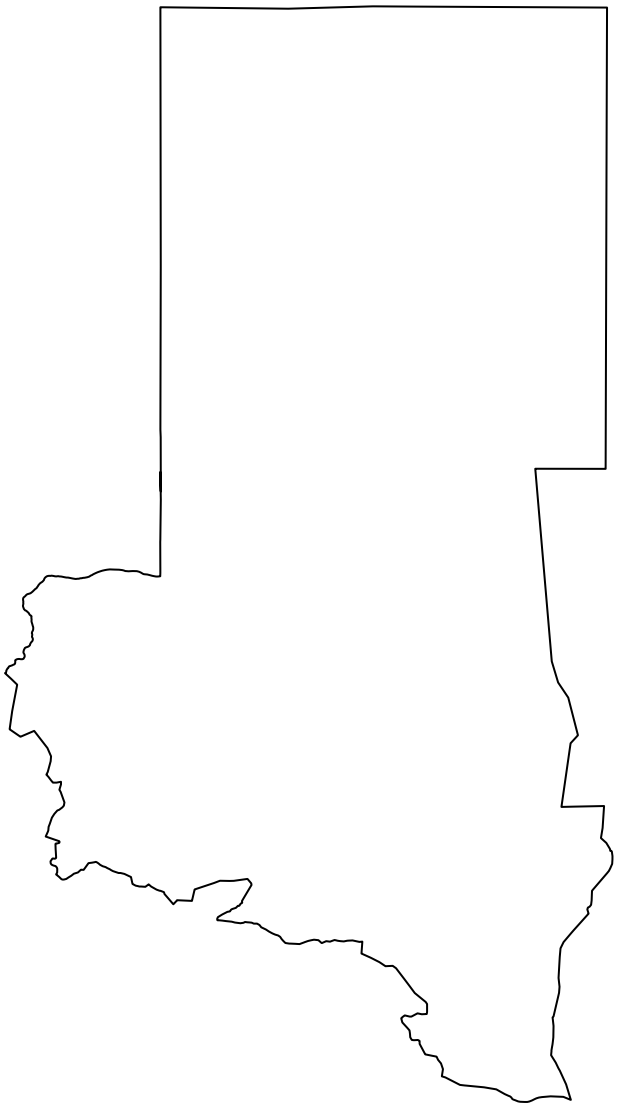


Figure 3.4: Map of North Darfur, Sudan

However, this map will look better if we can also show the nearby states and their boundaries. We can do this by plotting `sudan01` but this time specifying a `xlim` and `ylim` parameters using the `bbox()` of the map object for North Darfur. The following commands illustrate this:

```
nDarfur <- subset(sudan01, sudan01@data$State == "North Darfur")
ndLimits <- bbox(nDarfur)
plot(sudan01,
     xlim = ndLimits[1,],
     ylim = ndLimits[2,],
     border = "gray")
plot(nDarfur,
     xlim = ndLimits[1,],
     ylim = ndLimits[2,],
     border = "blue",
     add = TRUE)
```

The resulting map is:

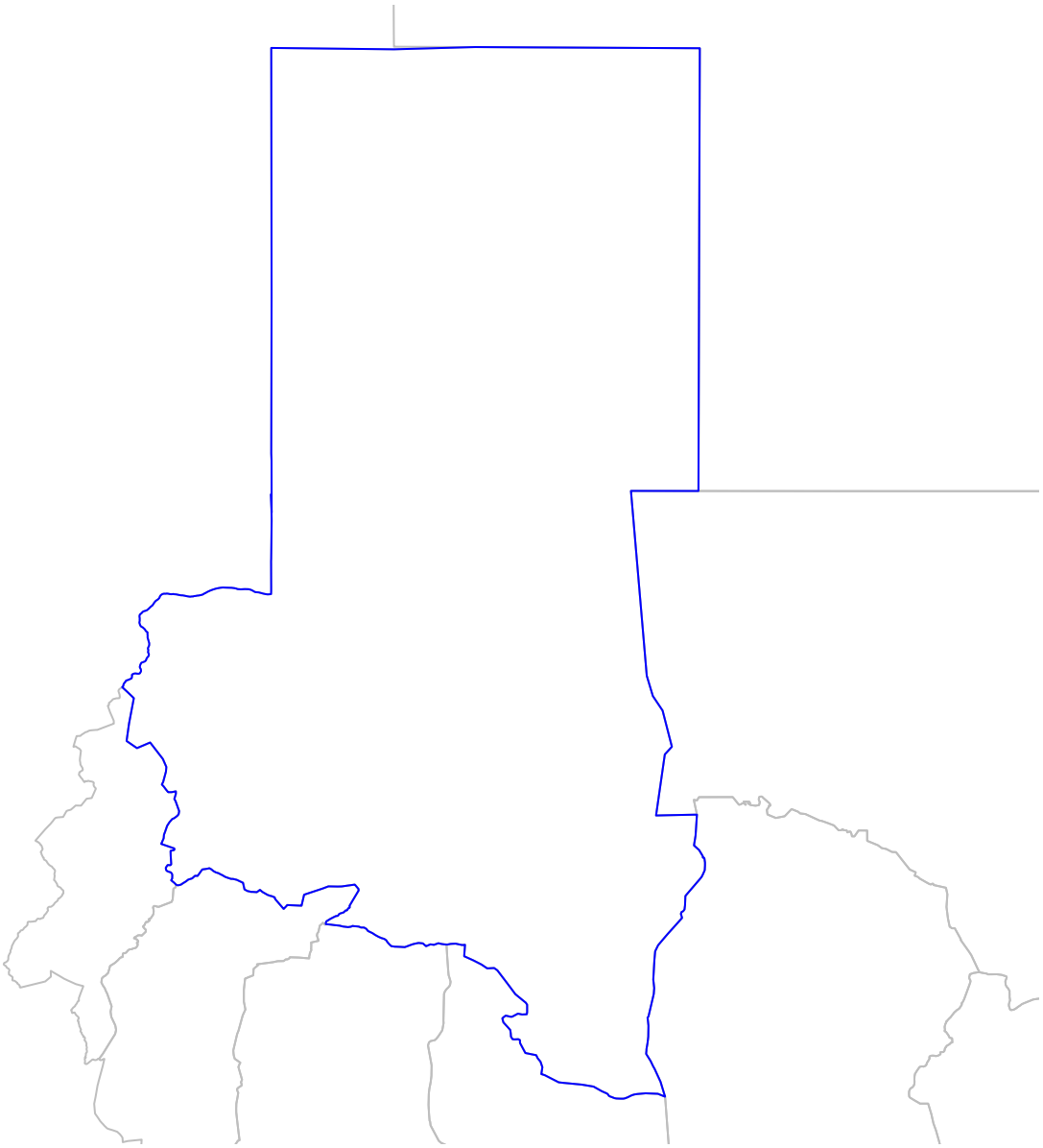


Figure 3.5: Map of North Darfur, Sudan

We can apply the same approach to get the map for each of the states. However, we can also create a looping function that will create and plot the maps for us automatically. This can be done as follows:

```
stateList <- as.vector(sudan01@data$State)

for(i in 1:nrow(sudan01@data)) {
  a <- subset(sudan01, sudan01@data$State == stateList[i])

  aLimits <- bbox(a)

  plot(sudan01,
       xlim = aLimits[1,],
       ylim = aLimits[2,],
       border = "gray")

  plot(a,
       xlim = aLimits[1,],
       ylim = aLimits[2,],
       border = "blue",
       add = TRUE)

  points(villages$X, villages$Y, pch = 20, cex = 0.3, col = "gray90")

  title(main = stateList[i], cex = 2)
}
```

The commands above will create a plot of the individual state maps on separate graphics device which can be saved one by one. An example of the maps is below.

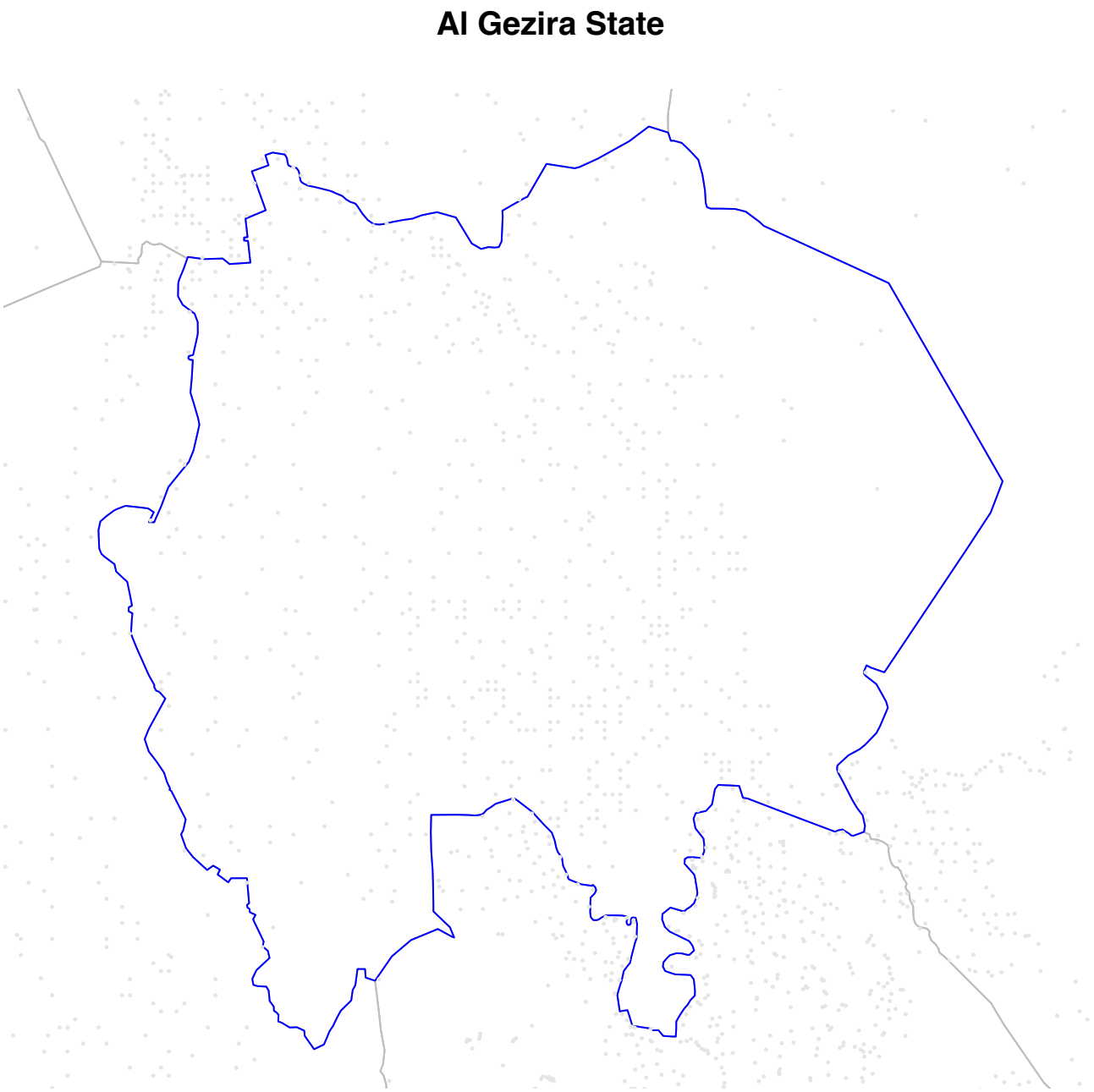


Figure 3.6: Map of Al Gezira State, Sudan

The other approach is to plot each of the individual state map on the same graphics device. This can be done using the following commands:

```
stateList <- as.vector(sudan01@data$State)

par(mar = c(2,2,2,2)); par(mfrow = c(6,3))

for(i in 1:nrow(sudan01@data)) {
  a <- subset(sudan01, sudan01@data$State == stateList[i])
  aLimits <- bbox(a)

  plot(sudan01,
       xlim = aLimits[1,],
       ylim = aLimits[2,],
       border = "gray")

  plot(a,
       xlim = aLimits[1,],
       ylim = aLimits[2,],
       lwd = 2,
       border = "blue",
       add = TRUE)

  points(x = villages$X,
         y = villages$Y,
         pch = 20,
         cex = 0.3,
         col = "gray90")

  title(main = paste(stateList[i], "State", sep = " "), cex = 2)
}
```

The resulting map is:

Figure 3.7: Map for each state of Sudan