3. Importing External Data

Often one of the first steps when doing a project in R is to import some data. This helpsheet will cover reading in a CSV file and a Shapefile. A CSV file is a basic format for data; a Shapefile is a collection of files that relate to geographic features (points, lines or polygons), associated attribute data and their projection information. Once such files have been read into R, you might need to tidy them up before doing any analysis - see helpsheet "2. Reworking and Recoding Data", for more information.

CSV Files

CSV (Comma Separated Values) files typically look like this when opened in a text editor:

```
colname1,colname2,....
row1value,row1value,....
row2value,row2value,....
```

Each column in separated by a comma, and each row with a carriage return. We will now read an example CSV file into a data frame in R. This is avaliable as a file (`example.csv`) which we will use in this exercise, and looks like:

```
Header text we want to ignore
Name,Age,Place,School
John,20,Liverpool,Hillside High School
Rachel,21,Norwich,Colman High School
Helen,34,Liverpool,Hillside High School
```

To read the file in, run this command:

```
# Set working directory
setwd("M:/R work")
# Read data from the web
file_location <- "http://data.alex-singleton.com/r-helpsheets/3/example.csv"
data <- read.csv(file_location, header = TRUE, skip = 1)
```

And to check that it has been input correctly, which is always a good idea with R, run:

```
data
```

This should output:

```
    Name Age     Place               School
1   John  20 Liverpool Hillside High School
2 Rachel  21   Norwich   Colman High School
3  Helen  34 Liverpool Hillside High School
```

Here, the object we created is called `"data"` and the function that we used is called `"read.csv"`, which has a number of options:

1. '`File_location`' is where the file is stored (within your working directory, see helpsheet 1. Basics for more details).

2. ‘`Header = TRUE`’ tells R that the CSV file has some header information (column names) in it, in this case `Name`, `Age`, `Place` and `School`.
3. ‘`Skip = 1`’ tells R to ignore the first line of the CSV file as we don’t want this in the data set. This was specified as `"Header text we want to ignore"` in the file.

We can now look at the object `data` in the normal way and, for example, check the column names using:

```
colnames(data)
```

Which should output:

```
[1] "Name"   "Age"    "Place"  "School"
```

If you want to rename columns or “recode” the attributes of your data, see helpsheet “2. Reworking and Recoding Data”.

Shapefiles

Shapefiles contain geographic data that we can also read into R, but to do this R needs some additional packages. These are already installed, but just need to be loaded.

To do this, run these commands:

```
library(sp)
library(rgeos)
library(maptools)
library(RColorBrewer)
library(GISTools)
library(rgdal)
```

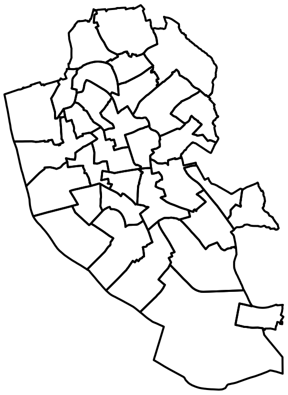
When you load each package, R will write some output to the console. Check for any error messages, and if everything seems to have worked, continue to the next section.

We can read in a Shapefile and then display it in R.

```
# Set working directory
setwd("M:/R work")
# Download data.zip from the web
download.file("http://data.alex-singleton.com/r-helpsheets/3/data.zip", "data.zip")
# Unzip file
unzip("data.zip")

# Read in Shapefile
Wards <- readOGR(".", "england-caswa_2001")

plot(Wards)
```



The object `Wards` now contains the attributes of the Shapefile. This has created a new type of object called a SpatialPolygonsDataFrame. If the Shapefile had been lines (e.g. roads), this would be a SpatialLinesDataFrame, or points, a SpatialPointsDataFrame. These new object types contain the spatial information (e.g. the boundary locations) as well as attribute data for each of the spatial features (e.g. Ward boundaries). The SpatialPolygonsDataFrame contains a number of different 'slots', each of which hold different information. Use the `slotNames` function to get a list of the different slots:

```
slotNames(Wards)
```

```
[1] "data"        "polygons"    "plotOrder"   "bbox"        "proj4string"
```

The slot `data` contains the attribute information for the shape file, and this is accessed using an @ symbol:

```
head(Wards@data)
```

```
   gid ons_label        name  label
0  545    00BYGC  St. Mary's 04BYGC
1 2003    00BYFN      Dingle 04BYFN
2 2007    00BYFU Grassendale 04BYFU
3 2008    00BYFC    Allerton 04BYFC
4 2010    00BYFG  Broadgreen 04BYFG
5 2015    00BYFS    Gillmoss 04BYFS
```

The data slot can be accessed in the same way as any standard data frame.