

7. Converting Coordinates

Sometimes you will need to convert spatial data from one coordinate system to another. This is often called reprojecting as different coordinate systems typically use different projections; i.e. the way in which the curved Earth is represented as a flat surface. There are lots of different projections, including the Mercator and Gall-Peters projections, as shown below:

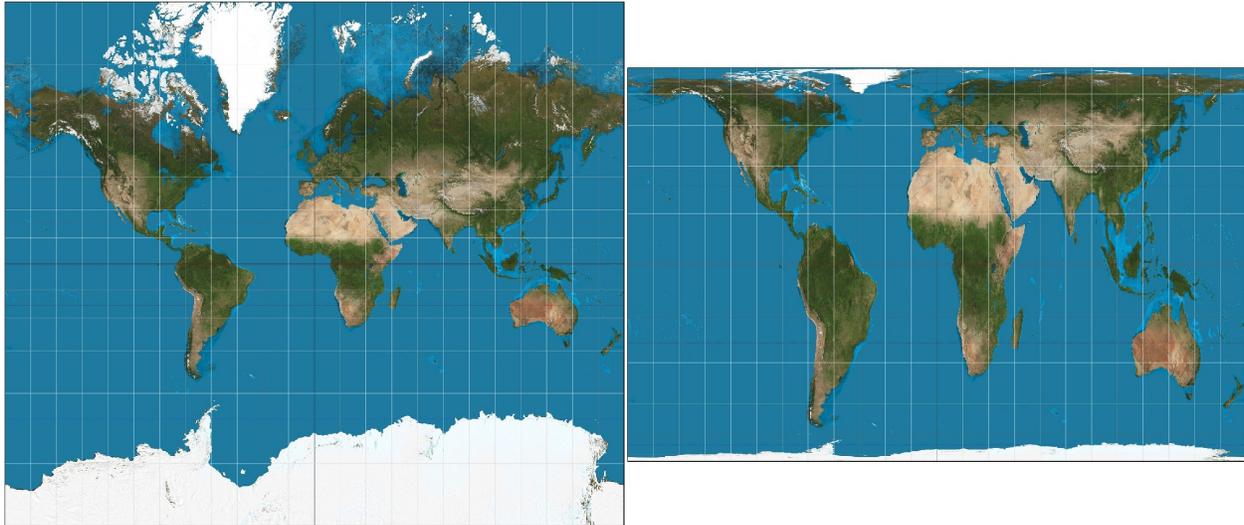


Figure 1: The Mercator projection on the left and the Gall-Peters projection on the right. Images from http://en.wikipedia.org/wiki/File:Mercator_projection_SW.jpg and http://en.wikipedia.org/wiki/File:Gall%E2%80%93Peters_projection_SW.jpg.

This helpsheet will take you through the process of converting BNG (British National Grid coordinates, Eastings and Northings) to Latitude and Longitude which requires reprojection between the OSBG36 and WGS84 datums. The same principle can be applied to any re-projection though.

Setup

There are some initial commands we need to run to setup R for this exercise. Firstly, loading the required library, and secondly, declaring some variables for the two different types of coordinate systems we will be using.

```
# Load the packages
library(rgdal)

# Variables for holding the coordinate system types (see:
# http://www.epsg.org/ for details)
ukgrid = "+init=epsg:27700"
latlong = "+init=epsg:4326"
```

We will use the locations of doctors surgeries data as an example. Download and import it using the following commands:

```
# Set working directory
setwd("M:/R work")
```

```

# Download data.zip from the web
download.file("http://data.alex-singleton.com/r-helpsheets/7/data.zip",
  "data.zip")

# Unzip file
unzip("data.zip")

# Get doctors surgeries data
GP <- read.csv("General Practices 2006.csv", header = TRUE, skip = 3)

# Extract the columns we want
GP <- subset(GP, select = c("Practice.Doctor.s.Name", "Easting",
  "Northing"))

# Rename the columns to something more helpful
colnames(GP) <- c("Surgery", "Easting", "Northing")

```

We now have the doctors surgeries, with their eastings and northings. To show a summary, run:

```
head(GP)
```

	Surgery	Easting	Northing
1	Alma Medical Centre	444214	519588
2	Melrose Surgery	446141	524086
3	Tithebarn Medical Centre	441957	521021
4	Barley Fields Medical Centre	444225	513852
5	Dr Banerjee	444592	519650
6	Dr Banerjee	442202	521234

We next need to convert the GP object from a data frame into a Spatial Data Frame.

```

# Remove those doctors surgeries with missing Eastings or
# Northings
GP <- subset(GP, Easting != "" | Northing != "")
# Create a unique ID for each GP
GP$GP_ID <- 1:nrow(GP)
# Create coordinates variable
coords <- cbind(Easting = as.numeric(as.character(GP$Easting)),
  Northing = as.numeric(as.character(GP$Northing)))
# Create the SpatialPointsDataFrame
GP_SP <- SpatialPointsDataFrame(coords, data = data.frame(GP$Surgery,
  GP$GP_ID), proj4string = CRS("+init=epsg:27700"))

```

GP_SP is now a spatial data frame. We can do a quick plot(GP_SP) to see what this looks like.

```

# Show the results
plot(GP_SP)

```



Because GP_SP is now a Spatial Data Frame, we need to use `head(GP_SP@data)` to view content.

```
head(GP_SP@data)
```

	GP.Surgery	GP.GP_ID
1	Alma Medical Centre	1
2	Melrose Surgery	2
3	Tithebarn Medical Centre	3
4	Barley Fields Medical Centre	4
5	Dr Banerjee	5
6	Dr Banerjee	6

You can see that the Eastings and Northings are no longer visible. In fact the eastings and northings are just in a different slot of the Spatial Data Frame. Try `head(GP_SP@coords)` instead.

```
head(GP_SP@coords)
```

	Easting	Northing
[1,]	444214	519588
[2,]	446141	524086
[3,]	441957	521021
[4,]	444225	513852
[5,]	444592	519650
[6,]	442202	521234

And there they are. The `Coords` slot will behave like a normal data frame, so we can access specific elements of it in the usual way, for example `head(GP_SP@coords[,1])`. See the helpsheet “1. R Basics” for more information on data frames.

Now, the command to reproject from British National Grid (Eastings and Northings) into WGS84 (Latitude and Longitude).

```
# Convert from Eastings and Northings to Latitude and Longitude
GP_SP_LL <- spTransform(GP_SP, CRS(latlong))
# we also need to rename the columns
colnames(GP_SP_LL@coords)[colnames(GP_SP_LL@coords) == "Easting"] <- "Longitude"
colnames(GP_SP_LL@coords)[colnames(GP_SP_LL@coords) == "Northing"] <- "Latitude"
```

```
head(GP_SP_LL@coords)
```

	Longitude	Latitude
[1,]	-1.318	54.57
[2,]	-1.287	54.61
[3,]	-1.352	54.58
[4,]	-1.318	54.52
[5,]	-1.312	54.57
[6,]	-1.349	54.58

Now the data are in Latitude and Longitude.