Assignment 2
(10%)

Instructor: Marie Auger-Méthé

Please send me before next class an e-mail with all the answers for this assignment. All R code should be in one script. Please indicate which lines are associated with each exercise by including comments in your code. I addition, send me a copy of the figures in your e-mail. You can save the figures in .pdf or .jpeg format.

Exercise 1

Create one SpatialLines object from three different files.

1. Import the m10_telemetry.json, m11_telemetry.json, and m12_telemetry.json files from the Ocean Tracking Network (OTN) Ocean Glider and Marine Observation project found at http://gliders.oceantrack.org/ajax. These are datasets associated with Slocum glider missions, more information on Slocum glider page: http://gliders.oceantrack.org/slocum.php. The Slocum datasets are in json files. There are many R packages that can import json data. For example, you can use the function fromJSON from the jsonlite package as follow: `slocum10 <- fromJSON(txt="m10_telemetry.json")`.
2. Use this data to create a single SpatialPointsDataFrame. The coordinates are found in the columns lon and lat, and the time associated with those locations are found in the column gpstime. This SpatialPointsDataFrame should have a column in the attributes that keeps track of the file from which the coordinates comes from (i.e. a column with m10 for each row associated with the m10_telemetry.json file and m11 for m11_telemetry.json, etc). **Tips:** You can create the column with the json file ID in each data.frame first (e.g., `slocum10$m <- "m10"`). You can then create one data.frame from multiple data.frames by using `rbind(slocum10, slocum11, slocum12)`.
3. Create a SpatialLines object for which each json file has its own Lines object and the ID of each of these Lines object should be the file identifier (e.g., m10).
4. Plot these SpatialLines. Each Lines object should be in a different colour.

Exercise 2

Create a SpatialLinesDataFrame with the SpatialLines object created in the previous exercise.

1. Make a `SpatialLinesDataFrame` from the `SpatialLines` object we've created in the previous examples. For the attributes, use the mean column values from each json file (e.g. `mean(slocum10$amphr)`). In particular, use the columns amphr and vacuum.
2. Plot the `SpatialLinesDataFrame` object and use the column amphr to display different colours.
3. Overlay on this plot, a plot of the `SpatialPointsDataFrame` that use a different colour for each json file.