Kodiak Rockfish Hydroaccoustic Data Analysis Using R

William Gaeuman

September 20, 2017

Contents

Introduction

This document describes the process for turning Echoview[1] accoustic rockfish data into estimates of station rockfish abundance and density using the graphical and statistical analysis software R. The first step involves creating a suitable directory stocked with the necessary input files. The second step involves invoking R and then loading and correctly running the required functions. This document additionally offers a peek under the hood for anyone so inclined.

Preliminaries

You can install the most recent version of R on your computer by visiting the R Project website at https://www.r-project.org. Once you have R on board, you will also need to install the R packages *zoo*, *tidyverse*, *rgdal* and *sp*. (A "package" is essentially a collection of interrelated functions and associated object definitions designed to perform a set of specific tasks.) To do this, open the **Packages** menu inside R and click on **Install package(s)...**. Unless you have done so previously, you will be prompted to select a **CRAN mirror** from

[1] A product of Echoview Software Pty Ltd

a pop-up list. Choose a location more or less nearby, e.g. NOT in Bulgaria, and click **OK**. Another pop-up list will appear. Find and select the desired package and click **OK**. Note that once a package has been installed, it must be loaded during each new R session for it to be generally available. This can be arranged either through the **Load package** option in the **Packages** menu or by using the R function `library()` with the package name, without quotes, as the argument.

Analysis Procedure

The rockfish data analysis proceeds station by station. The methods employed assume that the hydroacoustic data come from a single "track" that includes two or more sequential line transects across an individual rockfish station area. Both "grid" and "star" tracks are supported. The grid pattern ideally yields a set of equally spaced mutually parallel transects that are perpendicular to a fixed baseline. In star sampling, transects are traversed through a fixed common central point with equal angles between them like the spokes of a wheel. In theory, star transcect sampling should result in more efficient estimation of rockfish density and abundance when the distribution of fish is higly concentrated about the center of the pattern.

Creating The Station Directory

Given that your computer is equipped with R and the R packages *zoo*, *tidyverse*, *rgdal* and *sp*, the first step in the analysis is to create a station directory containing the following six files:

1. *track.csv*
2. *fish.csv*
3. *boundary.shp*
4. *boundary.shx*
5. *boundary.dbf*
6. *boundary.prj*

The first two files are **comma separated value files** that contain the accoustic data derived from the station track. Required covariates in the first file are ping time (hh:mm:ss.ss) and position coordinates in signed decimal degrees of latitude and longitude with corresponding column names *time*, *lat* and *lon*. Required covariates in the second file are these same three, also named *time*, *lat* and *lon*, plus *depth* expressed as a positive number in meters. The other four files are the `.shp, .shx, .dbf` and `.prj` components of the **shape file** that define the station area polygon. **Be advised that all of these name and file-type conventions are required.** It is also expected that the analyst understand and be familiar with the data in these files and have exercised some reasonable level of quality control on their content.

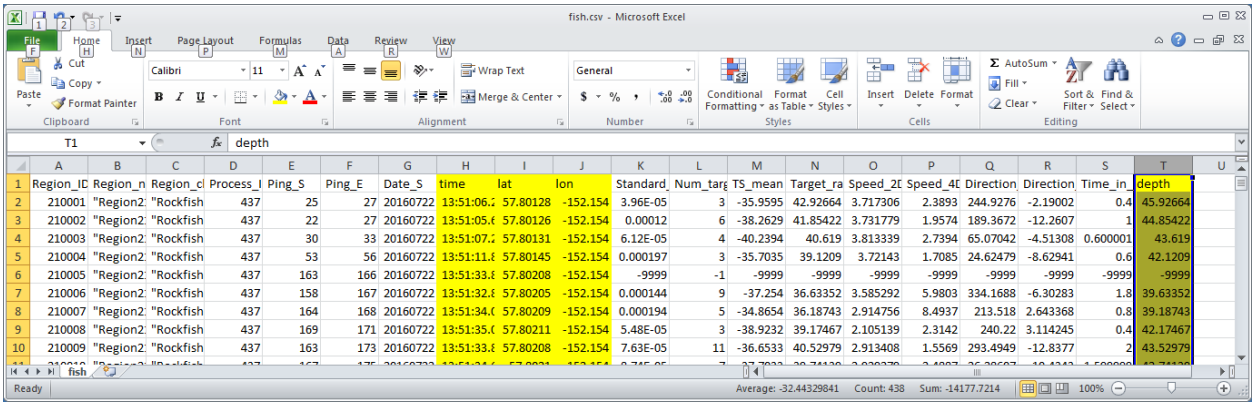


Figure 1: The four required columns of the *fish.csv* file.

Running The Analysis in R

The tools used to analyze the rockfish hydroacoustic data are stored in the R "workspace" file *rockfish.RData*, which must be loaded into the current R session to make them available. This can be done in R either by navigating to the file through the **Load Workspace** option in the **File** menu or by using the R function `load()`. Workspace contents can then be examined using the function `ls()`. On a Windows machine, for example, you would use

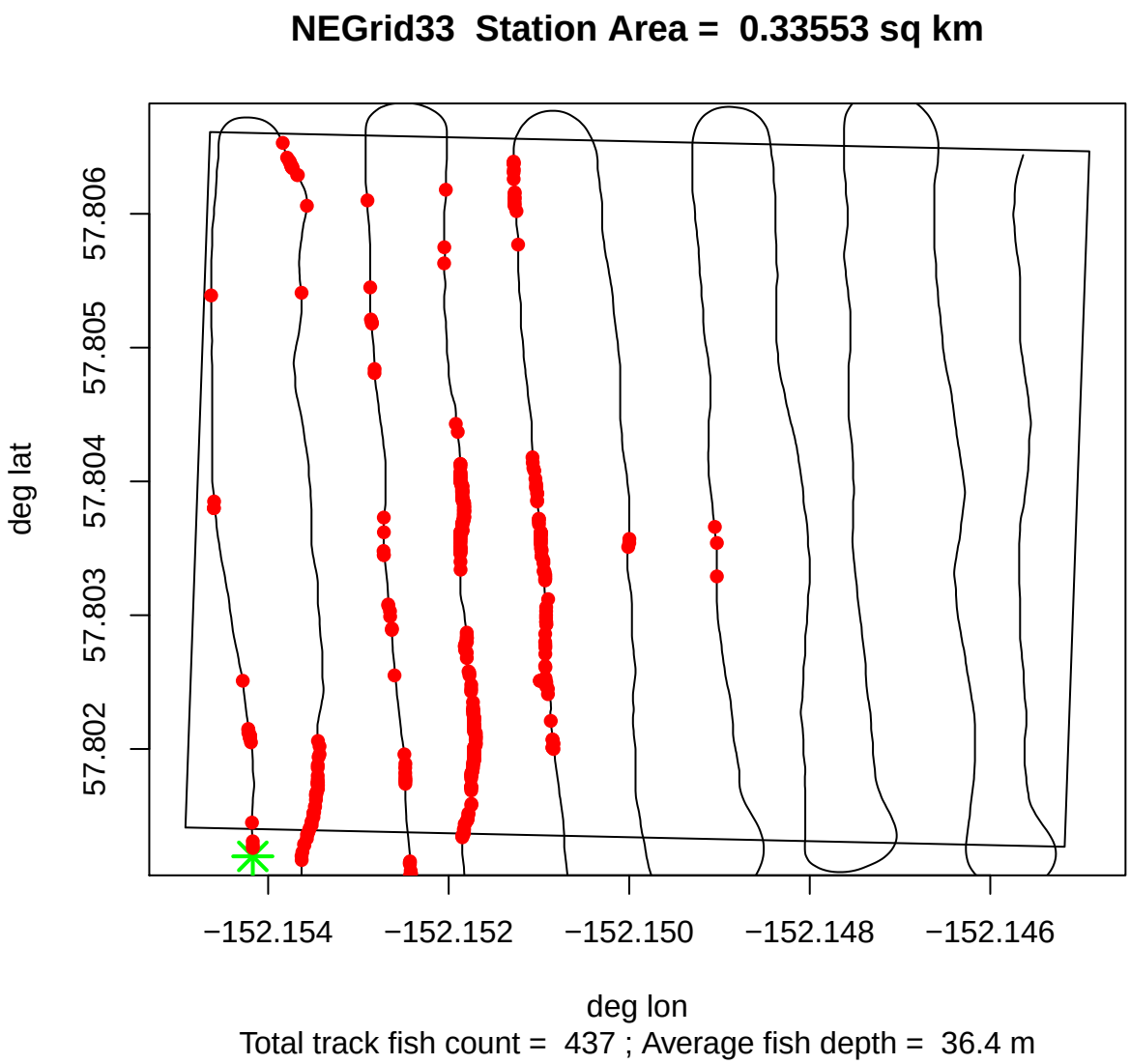
```
load("code/rockfish.RData")
ls()
```

to load the *rockfish.RData* workspace located in the *code* subdirectory of the current working directory and see a list of its contents, which currently consist of the three R functions `get_haver_distance()`, `display_station_data()` and `get_station_estimates()`.

The first function computes the great circle distance (km) between two points given in decimal degree latitude and longitude and is needed by the other two. The function `display_station_data()` plots the station track and area polygon with the positions of putative rockfish superimposed. Inputs are the station directory path name and station name and an optional `window` parameter specifying the the moving-average window size used to smooth track position coordinates, with default `window = 21`. The function `get_station_estimates()` computes station abundance and density estimates. Inputs to this function include those needed for function `display_station_data()`, along with the number of transects. The additional input `grid = FALSE` is required if star transect sampling was employed. Both functions automatically load the packages *zoo*, *tidyverse*, *rgdal* and *sp* provided these were previously installed; function execution will fail otherwise.

As an example, suppose the *rockfish.RData* file has already been loaded, as above, and that the required six files are located in a local subdirectory named *NEGrid33* associated with grid sampling of 11 transects at a station of the same name. Station data are displayed using the command

```
display_station_data("NEGrid33", "NEGrid33")
```



Station estimates of abundance and density may be obtained via the command

```
get_station_estimates("NEGrid33", "NEGrid33", 11)
```

which first displays the data so that the analyst can use the cursor to delineate the individual transects by clicking on the endpoints of each one where it intersects the station boundary. The order is irrelevant so long as all definining points—in this case `11 x 2 = 22`—are selected exactly once. Note that the plot window can be resized to facilite this process. The function then computes and returns the various estimates—for this example

```
#>    station    area obs.fish dens se.dens abund se.abund    cv
#> 1 NEGrid33 0.33553      400 5940    3223  1993     1081 0.543
```

These results are also written to the station directory as file *estimates.csv*.

A Quick Look Under The Hood

Estimation Methods

Implementation in R