Candidate Outliers

Notes on usage

Dr Stephen E. Lane[1]

[1]Chief Scientist, interadata

April 8, 2018

interadata
bespoke data science.

Contents

List of Figures

List of Code Examples

1. Introduction

This report provides instructions on how to screen incoming catch data for *candidate* outliers. We stress that any possible outliers should be labelled as candidate only; data that fall outside of 95% bounds (for example) may just be natural variation. Obvious mistakes in data entry — for example catch data an order of magnitude higher than normal — should be seen as outliers, and the root cause searched out[1].

2. Approach to Outlier Detection

We have chosen to use a meta-analytic approach to detecting candidate outliers at the diver-within-spatial management unit level. Using the 'metafor' package (Viechtbauer, 2010), we fit a random effects model to the diver catch data within each SMU.

A random effects model has been chosen due to the possibility of divers changing within SMU's. A random effects model adjusts the observed variation of each diver to account for the (essentially) random choice of diver within SMU.

The script provided to conduct the candidate outlier detection (`cpue-qa-reports.Rmd`) performs calculations based on the most recent year of catch data entered into a data spreadsheet. By default, it is set up to use the supplied raw catch effort data `RawCatchEffortUnfiltered_1979_2016_DiverPFN.xlsx` (hereafter, CPUE spreadsheet). The most recent year of data is used based on the assumption that all previous years have had candidate outliers reviewed, and adjusted if necessary.

3. Method

There are two key scripts for the candidate outlier detection:

- `cpue-qa-reports.Rmd` is a parameterised R Markdown report that performs the calculations for a given SMU, and creates a HTML report; and
- `create-outlier-reports.R` is a driver script that creates the reports for a list of SMU's.

A number of packages (and their dependencies) are required to create the reports; these are shown in Code Chunk 3.1.

R Chunk 3.1.

```
## If these packages are not installed, install them with:
## install.packages("package_name")
library(here)
library(rmarkdown)
library(dplyr)
library(metafor)
library(readxl)
```

The `cpue-qa-reports.Rmd` parameterised reports are saved in the reports directory by default. If this does not exist, the `create-outlier-reports.R` script will

[1] As suggested here, this will most often be due to a data entry error.

create it. We use R package `here` (Müller, 2017) to locate files within the VFA directory. `here` looks for an R Studio project file (in this case, it is called `vfa.Rproj`) and then creates absolute references to files based on the user's operating system. For details, see the help file (`?here`).

In order to detect candidate outlier divers, each individual diver has their individual dives turned into catch per unit effort (CPUE) by summing both their Blacklip and Greenlip abalone catches (if they have both) and dividing by their effort. These variables are all recorded in the CPUE spreadsheet provided. If divers have not recorded their effort data, or the effort data is missing, they are removed from further analysis. Similarly, if a diver has had only one dive, they are removed from analysis. Each of these removals is documented in the produced reports.

Each diver has their CPUE from their dives summarised by the mean and standard deviation, which are required for the candidate outlier detection. These summaries form the input into the `rma` routine within the `metafor` package.

4. Creating the Reports

We now demonstrate how to generate the candidate outlier reports. Firstly, we assume a directory structure similar to that shown in Figure 1. `cpue-qa-reports.Rmd` should be in the **Rmd** folder, `create-outlier-reports.R` in the **R** folder, and the CPUE spreadsheet in the **data-raw** folder.

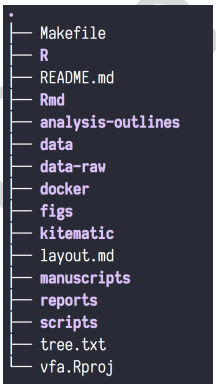

Figure 1: Top-level directory structure required for creating reports.

Code Chunk 4.1 shows how to compile a report for a single SMU, in this example, Airport. Note the use of the `here` command, which as described earlier, provides intelligent folder sourcing. The compiled report will be found in the **reports** folder, and can be opened with any web browser. The report can also be compiled to a Word document, however some styling is lost; for an example of how to compile to Word, see Code Chunk A.1.

R Chunk 4.1.

```
library(rmarkdown)
library(here)
render(input = here("Rmd", "cpue-qa-reports.Rmd"),
       output_file = here("reports", "candidate_outliers_airport.html"),
       params = list(
           smu = "Airport",
           input_data = here("data-raw",
                             "RawCatchEffortUnfiltered_1979_2016_DiverPFN.xlsx")))
```

An R script (`create-outlier-reports.R`) has been provided to compile reports for all SMU's. This script relies on a list of all the SMU's present in the most recent year's worth of data. As an example, Code Chunk 4.2 shows the current status of this list. This list of SMU's *must be exact* for the reporting to work; if the list does change, the list in `create-outlier-reports.R` must be changed appropriately. Code Chunk 4.3 shows how to create the reports for each listed SMU; the corresponding commands to produce Word reports is shown in Code Chunk A.2.

R Chunk 4.2.

```
SMU <- c('Airport', 'BACK BEACHES', 'CAPE LIPTRAP', 'CAPE OTWAY',
         'CLIFFY GROUP', 'FLINDERS', 'Julia Percy Island', 'KILCUNDA',
         'Mallacoota Central', 'Mallacoota East', 'Mallacoota Large',
         'Mallacoota Small', 'Mallacoota West', 'Marlo', 'PHILLIP ISLAND',
         'Port Fairy', 'Portland', 'PROM EASTSIDE', 'PROM WESTSIDE',
         'SHIPWRECK COAST', 'SURFCOAST', 'Warrnambool')
```

R Chunk 4.3.

```
library(here)
source(here("R", "create-outlier-reports.R"))
sapply(SMU, create_report, datafile = data_file)
```

A sample of candidate outlier reports for 2016 is provided in the `reports/` directory in the provided materials.

References

Müller, Kirill (2017). *here: A Simpler Way to Find Your Files*. R package version 0.1.
Viechtbauer, Wolfgang (2010). "Conducting meta-analyses in R with the metafor package". In: *Journal of Statistical Software* 36.3, pp. 1–48.

A. Extra Compilation Options

This appendix shows code to compile the reports into Word format. Some styling is lost using this format, so we do not recommend this.

Code Chunk A.1 shows how to compile a single candidate outlier report (for the Airport SMU) into Word format. Code Chunk A.2 shows how to compile candidate outlier reports for all SMU's.

R Chunk A.1.

```
library(rmarkdown)
library(here)
render(input = here("Rmd", "cpue-qa-reports.Rmd"),
       output_file = here("reports", "candidate_outliers_airport.docx"),
       output_format = "word_document",
       params = list(
           smu = "Airport",
           input_data = here("data-raw",
                             "RawCatchEffortUnfiltered_1979_2016_DiverPFN.xlsx")))
```

R Chunk A.2.

```
library(here)
source(here("R", "create-outlier-reports.R"))
sapply(SMU, create_report_word, datafile = data_file)
```