Catch per Unit Effort Standardisation

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 details on the methodology used to create standardised catch per unit effort (CPUE) estimates, and the infrastructure developed to produce reports of model fitting to each of the spatial management units (SMU's) of interest. We outline the methodology in Section 2, and provide implementation details in Sections 3 and 4.

2. Standardisation Methodology

One of the primary goals in this standardisation is to provide consistent CPUE inputs into the abalone harvest strategy. This is performed on a yearly basis, and so we feel the best approach for the standardisation is to also focus on a year-by-year basis — thus we aggregate all records to a yearly basis, keeping the groupings of year/reef/diver.

Aggregation to the yearly level will provide precision in the models, by removing the source of uncertainty at the individual dive level. For example, uncertainty due to weather conditions on the day of the dive. This is appropriate as it focuses on the goal of the standardisation.

To perform the standardisation, we fit a sequence of regression models to the total catch (kg) per diver on each reef within the SMU over all recent years. That is, the models are fit within SMU's. Due to the restriction that catch must be greater than 0, we use a Gamma distribution for the response variable. Categorical variables are modelled using probability distributions. We use an offset adjustment for the effort spent on the catch to account for uneven effort between divers and/or reefs.

Equations (1)–(Model 3) show the outcome/regression model for the total catch, and the sequence of models used to adjust for year, reef and diver effects.

$$C_{trd} \sim \text{Gamma}(\alpha, \beta_{trd}) \tag{1}$$

$$\beta_{trd} = \mu + E_{trd} + \text{Year}_t + \text{Reef}_r + \text{Diver}_d \tag{Model 1}$$

$$\beta_{trd} = \mu + E_{trd} + \text{Year}_t + \text{Reef}_r + \text{Diver}_d + \text{Year by Reef}_{tr} \tag{Model 2}$$

$$\beta_{trd} = \mu + E_{trd} + \text{Year}_t + \text{Reef}_r + \text{Diver}_d + \text{Year by Reef}_{tr} + \text{Year by Diver}_{td} \tag{Model 3}$$

where C_{trd} and E_{trd} are the total catch and effort recorded by diver d on reef r in year t. Year_t is the effect of year t, Reef_r the effect of reef r, and Diver_d the effect of diver d, with similarly named interaction effects.

There is a slight hitch with the Cliffy Group SMU, as there is only a single reef recorded in the data. We still fit a similar group of models as Equations (Model 1)–(Model 3), but remove any terms involving reef effects.

To complete the specification, the shape parameter α, along with the year, reef, diver and associated interaction terms, are given vague priors which are detailed in Appendix A.

All models are fit using the `rstan` package (Stan Development Team, 2017), which is a general purpose Monte Carlo simulation package that uses a very efficient Hamiltonian Monte Carlo approach. For more details, see the `rstan` website, and associated documentation.

3. Infrastructure for the Standardisation

In this section we detail the infrastructure to implement the standardisation for each SMU. We make special note here that due to the methods used (Section 2), the implementation can be computationally expensive. As an example, calculating the standardisation for a single SMU may take up to an hour.

We have strived to make the implementation as robust as possible. This means that models are refit if certain errors are detected. With such a generalised brief, it is impossible to take account of all issues that may occur with the modelling, and so our implementation focuses on two main control parameters: the number of iterations that ended with divergent transitions, and the number of iterations required to achieve a certain precision in the Markov chains. It is beyond the scope of this report to go into detail of these control parameters, but we refer to the `fitAndCheckModels.R` functions as well as the `rstan` documentation for further details. If errors do eventuate, the CPUE reports will state this.

We have provided a script to produce the reports in a parameterised fashion (`cpue-standardisation-reports.Rmd`), similar to the scripts provided for the candidate outliers reports[1]. These scripts have been set up to use the supplied raw catch effort data `RawCatchEffortUnfiltered_1979_2016_DiverPFN.xlsx` (hereafter, CPUE spreadsheet).

There are two key scripts to produce the standardisation:

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

There are a number of other scripts, functions and model definitions. These are provided in the `R/` and `stan/` directories, and are called and used by the two key scripts detailed above.

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)
library(ggplot2)
library(forcats)
library(tidyr)
library(rstan)
library(loo)
## For quicker computation, is recommend, but not required
library(parallel)
library(lubridate)
```

[1] See `candidate-outliers.pdf`

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

Some preprocessing is performed in the `create-models.R` script: each individual diver has their individual dives summed by year and reef variables. Total catch is calculated by summing both their Blacklip and Greenlip abalone catches[2] (if they have both); total effort is calculated by summing their Effort data. 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.

4. Creating the Reports

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

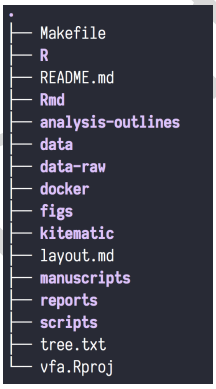

Figure 1: Top-level directory structure required for creating reports.

The reports can be fit using the `create-models.R` script. We will step through this script, detailing what it is doing in the following code chunks. Code Chunk 4.1 loads all of the required libraries, along with the associated driver scripts (as mentioned in Section 3). Some points to note here: we use the `parallel` package to implement multicore processing — this will speed up the fitting of the models. The option provide (`mc.cores`) tells `rstan` to use multiple cores when possible. If you can't use parallel processing, don't load the `parallel` library, and don't set that option. The `theme_set` command sets graphics niceties for the `ggplot2` package.

[2]We need to use both Blacklip and Greenlip catch volumes, as the effort data is only recorded for both catches combined.

R Chunk 4.1.

```
library(here)
library(readxl)
library(dplyr)
library(tidyr)
library(parallel)
library(rstan)
library(loo)
library(ggplot2)
library(rmarkdown)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores() - 2)

## Load required scripts
source(here("R", "createStanData.R"))
source(here("R", "stan-helpers.R"))
source(here("R", "fitAndCheckModels.R"))
source(here("R", "themeVFA.R"))
theme_set(themeVFA(16))
```

Code Chunk 4.2 loads in the CPUE spreadsheet, and sets the list of SMU's for which standardised CPUE is required.

R Chunk 4.2.

```
## Load data
data_file <- read_excel(
    here("data-raw", "RawCatchEffortUnfiltered_1979_2016_DiverPFN.xlsx")
)
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')

## Create reports/data directory for modelling outputs it it doesn't exist.
if(!dir.exists(here("data"))) dir.create(here("data"))
if(!dir.exists(here("reports"))) dir.create(here("reports"))
```

The first part of Code Chunk 4.3 shows how to produce a standardised CPUE report for a single SMU, namely Airport. In the second part of Code Chunk 4.3, we show how to loop over all the SMU's to produce a series of reports. As discussed, a single SMU can be very computationally intensive — using the `sapply` command to loop over all the SMU's is likely to take in excess of 24 hours to complete.

R Chunk 4.3.

```
## Fit models (restricted to years 2000 and above)
## Due to the types of models, this can be quite time consuming.
min_year <- 2000
max_year <- max(data_file$QuotaYear)
fitMod(
    "Airport",
    data_file = data_file,
    min_year = min_year,
    max_year = max_year
)

sapply(SMU, fitMod, data_file = data_file, min_year = min_year,
       max_year = max_year)
```

A sample of CPUE standardisation reports for the period 2000 to 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.
Stan Development Team (2017). *RStan: the R interface to Stan*. R package version 2.17.2.

A. Prior Specifications

This appendix provides details on the prior specifications for the models in Section 2. Each of the group effects is modelled independently, and given vague Student t priors, with Cauchy priors for the standard deviations:

$$
\begin{aligned}
\mu &\sim \text{Normal}(0, 5) \\
\text{Year}_t &\sim \text{Student t}_3(0, \sigma_Y) \\
\text{Reef}_r &\sim \text{Student t}_3(0, \sigma_R) \\
\text{Diver}_d &\sim \text{Student t}_3(0, \sigma_D) \\
\text{Year by Reef}_{tr} &\sim \text{Student t}_3(0, \sigma_{YR}) \\
\text{Year by Diver}_{td} &\sim \text{Student t}_3(0, \sigma_{YD}) \\
\sigma_Y &\sim \text{Cauchy}(0, 2.5) \\
\sigma_R &\sim \text{Cauchy}(0, 2.5) \\
\sigma_D &\sim \text{Cauchy}(0, 2.5) \\
\sigma_{YR} &\sim \text{Cauchy}(0, 2.5) \\
\sigma_{YD} &\sim \text{Cauchy}(0, 2.5)
\end{aligned}
$$