sa4ss

Chantel Wetzel

2021-02-06

Getting Started

Want to create an assessment document using **sa4ss**?

Load up the package available on github:

```
remotes::install_github("nwfsc-assess/sa4ss")
library(sa4ss)
```

Now that the package is installed in R, let's create a document.

Step 1

Set the working directory. The document template will be created in the defined location.

```
setwd("C:/Assessments/2021/copper_rockfish_2021/write_up/wa")
```

Step 2

Call the draft function to create a general document with species, authors, and title created:

```
sa4ss::draft(authors = c("Chantel R. Wetzel",
                         "Brian J. Langseth",
                         "Jason M. Cope",
                         "Theresa Tsou",
                         "Kristen E. Hinton"),
             species = "Copper Rockfish",
             latin = "Sebastes caurinus",
             coast = "Washington US West",
             type = c("sa"),
             create_dir = FALSE,
             edit = FALSE)
```

The above example will automatically populate a document for copper rockfish with the above authors and their associated affiliation. **sa4ss* has a list of common authors and their affiliations already set-up. In order for authors to be found in the available list and their affiliation the names need to be entered exactly as expected here: https://github.com/nwfsc-assess/sa4ss/blob/master/data-raw/authors.csv.

The above call creates a number of files that will be used in the document (Introduction, Data, . . .) and their subsections. All files have a leading number at the start of the Rmd file names. The order of the numbers indicates the order that each separate file will be stitched together to create the whole document. If you would like to move the order of a text section of the document, you will just need to adjust the number order. Additionally, when the document is drafted a standard set of sections are created, however, not every assessment will need each section. A common example are the standard survey files created (e.g. '21s-wcgbts'). If your assessment will not include data from a specific survey, this file can be removed from the document by deleting the file.

Step 3

Create an Rdata object based upon the base model. The base model can be located anywhere on your machine.

```r
sa4ss::read_model(mod_loc = "C:/Assessments/2021/copper_rockfish_2021/models/base")
```

This will create an Rdata object called 00mod.Rdata in your working directory.

Step 4

Render the pdf document based on the draft template created:

```r
bookdown::render_book("00a.Rmd", clean = FALSE, output_dir = getwd())
```

This will create the pdf document which will be saved directly in the working directory. However, one may prefer to have the pdf saved in a specific folder. One default approach is to call the render_book function without specifying the output_dir which will put the pdf document in a separate folder in the working directory.

```r
bookdown::render_book("00a.Rmd", clean = FALSE)
```

Editing the Document

A number of documents in the specified directory should not be available. These documents correspond to different sections required by the Groundfish Terms of Reference.

When editing make sure to render the document on a regular basis by using either of the above calls to:

```r
bookdown::render_book("00a.Rmd", clean = FALSE, output_dir = getwd())
```

This will allow users to identify errors quickly. If the document rendered correctly the user should see that "Output created: _main.pdf" followed by miscellaneous other warnings in the R terminal.

Returning to the Document

If you are returning to edit your document at a later point, you will not need to redo steps 1-3 above. To begin working on your document again you should:

1. Open an new R terminal.
2. Load of the package **library(sa4ss)**.

3. Set the working directory to where your document is located.
4. Open Rmd files and start editing your document. To render the book at any point in this process you just need to run the following call:

```
bookdown::render_book("00a.Rmd", clean = FALSE)
```

Error in Rendering

If you see an error and the document fails to render you will want to:

1. Open the _main.tex file, read the error message in the R terminal, and navigate to that line in the _main.tex file.

2. Correct the error in the Rmd files.
3. Delete the _main.Rmd file in the folder.
4. Rerun the render line of code:

```
bookdown::render_book("00a.Rmd", clean = FALSE, output_dir = getwd())
```

Updating the Base Model

If the base model changes, the user will need to rerun the **read_model** function to load in the output from the new model run and locate and/or create figures and tables

```
sa4ss::read_model(mod_loc = "C:/Assessments/2021/copper_rockfish_2021/models/new_base",
                    create_plots = TRUE,
                          save_loc = file.path(getwd(), "tex_tables"),
                          verbose = TRUE))
```

This will create a new model output Rdata file containing the output from **r4ss::SS_output** function which will replace the old file and be used to update the document to the new model the next time the document is rendered. This file is loaded automatically (sourced in the 00.Rmd file). In addition to creating an Rdata object with model information the **sa4ss::read_model** function also calls the **r4ss::SSexecutivesummary** function and creates csv tables used in the Executive Summary section and LaTex tex tables which can be used directly in the document.

Tips & Tricks

Share Text Across Documents

If you are creating an area-specific assessment document for species you may want to use the same text across multiple documents. This can be easily done by creating separate folder for sections of text that will be shared across documents. Text should be included in a file with an Rmd extension and can be called using a **R** code chunk with the **knit_child** command in the **knitr** package:

```
other_loc = file.path('folder', 'child_example.Rmd')
res <- knitr::knit_child(other_loc, quiet = TRUE)
cat(res, sep = '\n')
#>
#>
#> Here is the text in the child document.
```

Add Figures from r4ss

There are a number of ways to add figures to the document, either in the executive summary or figures sections. The first approach is to directly specify where the document is located, add a figure caption and alternative text caption (for accessibility), and specify a figure label which allows the user to reference the figure in the document. In the below example the figure (data-plot.png) is located in the same directory as the Rmd files being rendered. No **R** code chunk is needed in this case.

```
# This is the text that is placed in the document:

![Summary of data sources used in the base model.\label{fig:data-plot}](data-plot.png)
{width=100% height=100% alt="Summary of data sources used in the base model"}
```

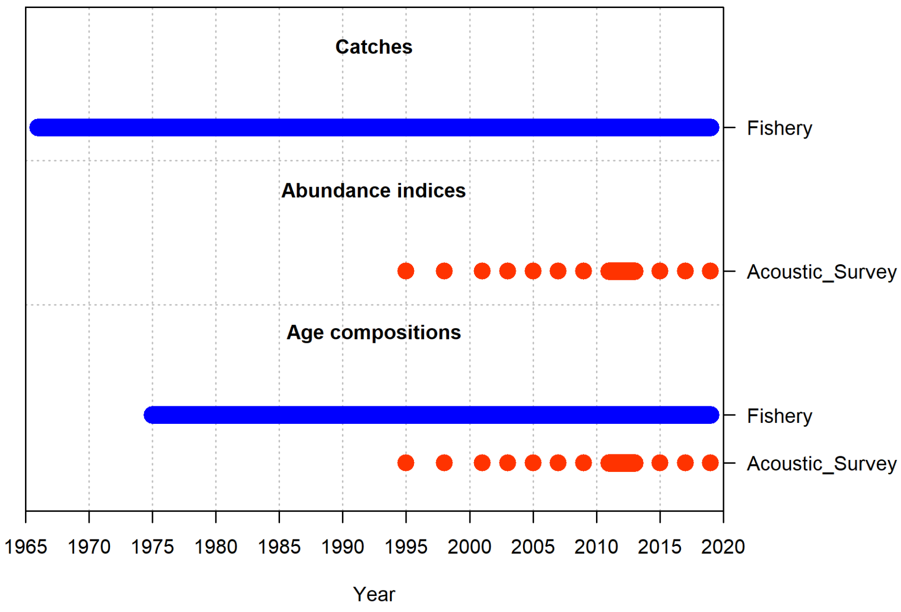

Figure 1: Summary of data sources used in the base model.

Alternatively, the figure you want to include may be located in a folder outside the folder that you are rendering the document within. To do this easily a function has been added to the **sa4ss** package that allows users to do this within a **R** code chunk.

```
# The add_figure function has the user specific the figure and its location to be used.

add_figure(
filein = file.path(mod_loc, "plots", "data_plot.png"),
caption = "Summary of data sources used in the base model",
label = 'data-plot')
```

The above function, **sa4ss::add_figure**, will set the alternative text equal to the caption unless specifically specified by the user.

Both examples above allow the user to reference the figure in the text inline by:

```
\ref{fig:data-plot}
```

which will look like this: Figure 1.

Adding Tables to the Document

There are two commonly used options to add tables to the document: 1) create tables directly using **R** code blocks or 2) users can also load existing tex table files.

Option 1: Using Code Chunks Directly

Another option for creating a table in a **sa4ss** document is to use a snippet of **R** code. The below example reads in an existing csv and creates the needed format for *sa4ss**:

```
tab = read.csv(file.path(mod_loc, 'tables', 'data_weights.csv'))
col_names = c('Method',
                    'Commercial Lengths',
              'Recreational Lengths')

sa4ss::table_format(x = tab,
             caption = "Data weights applied by each alternative data weighting methods.",
             label = "dw",
             align = 'l',
             col_names = col_names)
```

where the **sa4ss::table_format** function allows you to specify the table caption to use, the column names (col_names), and how to reference the table in the text (label). This table will appear in the document as:

Table 1: Data weights applied by each alternative data weighting methods.

Method	Commercial Lengths	Recreational Lengths
Francis	0.4544	0.1172
McAllister-Ianelli	0.6730	0.1010
Dirichlet Multinomial	0.8330	0.6130

R code chunks can also create tables from **R** objects that are loading in the work space. One item that will be loaded into your work session is a Rdata object with the output from **r4ss::SS_output** for your model. This Rdata object is created using the **sa4ss::read_model** function (see “Updating the Base Model” section above for information).

```r
like = cbind(rownames(model$likelihoods_used),  model$likelihoods_used$values)
like[,1] = gsub("\\_", " ", like[,1])
colnames(like) = c("Label", "Total")

sa4ss::table_format(x = like,
             caption = "Likelihoods in the model",
             label = "likelihood",
             align = 'l')
```

The above when executed in the document will create the following table:

Table 2: Likelihoods in the model

Label	Total
TOTAL	222.158
Catch	6.82562e-08
Equil catch	0
Length comp	215.04
Recruitment	7.10851
InitEQ Regime	0
Forecast Recruitment	0.00808815
Parm priors	2.51082e-09
Parm softbounds	0.001829
Parm devs	0
Crash Pen	0

Option 2: Using Tex Files

The **sa4ss::read_model** will automatically create tex files from csv files created from the **r4ss::SSexecutivesummary** function and save those tex files where you specify. The following example tells to function to read the base model (specified by mod_loc), create the **r4ss::SS_plots** from the model, create the **r4ss::SSexecutivesummary** tables and save the converted tex files in the directory specified by save_loc. The working directory should already be set to the location of your assessment document you are building.

```r
sa4ss::read_model(mod_loc = "C:/Assessments/2021/copper_rockfish_2021/models/new_base",
                    create_plots = TRUE,
                          save_loc = file.path(getwd(), "tex_tables"),
                          verbose = TRUE))
```

Once the tex files have been created they can be sourced into your assessment document using the **include** command in LaTex:

```
\include{tex_tables/Catches_All_Years.tex}
```

Users often need to add tables describing data or other items that are not included in the suite of tables created by the **r4ss::SSexecutivesummary** function. One example would be a table of the number of trips and lengths sampled per year for a fleet in your model. Users could to this via Option 1 described above or they could create a tex file for the existing csv containing the needed information. To do this, you will need to create a master csv file that specifies specific attributes for the various csv files you would like to include in your document. The master csv file will need to have these specific columns with the exact name and order as: "caption", "altcaption" "label", "filename", "loc". In the "caption" column you will type out the table caption, "altcaption" is the text for the alternative 508 compliance caption (a value of NA will default to use the caption), "label" is the way the table will be referenced in the document, "filename" is the name of csv file to read, and "loc" is the location of the csv file. Once this master csv file is created you can create tex files for all items using the **sa4ss::es_table_tex** function:

```
es_table_tex(dir = file.path(getwd(), 'tables'),
            save_loc = file.path(getwd(), "tex_tables"),
            csv_name = "all_tables.csv")
```

where dir specifies the location of the master csv file, the save_loc is where the newly created tex files will be saved, and csv_name is the name of the master csv file.

Referencing Figures and Tables

Tables and figures can be referenced in text by using the ref command:

```
The data weights applied in the base model are shown in Table \ref{tab:dw}.

The data used in the model are shown in Figure \ref{fig:data-plot}.
```

Each of the above examples will appear in the text as:

The data weights applied in the base model are shown in Table 1.

The data used in the model are shown in Figure 1.

Create or Load Specific R Objects

This can be done in any of the Rmd document files, however, I typically prefer to do this is a specific location to keep things organized. The "00a.Rmd" file already has an **R** code chuck where some standard variables are set. I like to add items here that I will then use across other files:

```
####  Standard Items Defined by sa4ss
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)
knitr::knit_hooks$set(plot = function(x,options) {
      base = knitr::opts_knit$get('base.url')
      if (is.null(base)) base = ''
      alt = ifelse (is.null(options$alt),"",options$alt)
      cap = ifelse (is.null(options$caption),"",options$caption)
      if (alt != ""){
        sprintf('![%s](%s%s "%s")', cap, base, x, alt)
      } else {
        sprintf('![%s](%s%s)', cap, base, x)
        }
  })
```

```
load("00opts.Rdata")

# I create the species name in a way I like to use throughout the docuemnt
spp = 'copper rockfish'
Spp = 'Copper rockfish'

# Here is a specific location where many of my exploratory
# figures are located that I want to include
data_fig_loc =
"C://Assessments/2021/copper_rockfish_2021/data/biology/plots"

# Here is where your model information created by the sa4ss::read_model
# function is loaded into the workspace
if(file.exists("00mod.Rdata")){
  load("00mod.Rdata")
} else {
  print("Model output not being read. Please run the read_model() function.")
}
```

Text and R Objects Inline

I will often use items defined in the "00a.Rmd" file when writing in my document.

```
`r Spp` range along the U.S. West Coast ranging into the Canadian waters.
```

which will appear in the text as:

Copper rockfish range along the U.S. West Coast ranging into the Canadian waters.

This can be really handy for incorporating model estimates into text descriptions.

```
The base model assumed a steepness value of
`r round(model$parameters[model$parameters$Label == "SR_BH_steep","Value"],2)`.
```

which will appear in the text as:

The base model assumed a steepness value of 0.72.