NAFOdown: An R package for Creating NAFO Scientific Council Documents

NAFOdown is an R package designed to simplify the process of creating and updating NAFO documents, such as SCRs and STACFIS reports. The package utilizes the [bookdown](https://bookdown.org/yihui/bookdown/) package which facilitates the integration of Markdown syntax and R code and, as such, effectively provides a “one-stop-shop” tool through which text, analyses, plots and tables can be written in parallel and knit into a stand-alone and reproducible document. Such a workflow minimizes the significant amount of manual effort associated with coping and pasting data, summary statistics and plots from one program to another. This approach also minimizes the amount of time spent on formatting documents as the NAFOdown package, like the [csasdown](https://github.com/pbs-assess/csasdown/) and [thesisdown](https://github.com/ismayc/thesisdown/) packages which this package is based on, includes templates that conform to formatting guidelines. Specifically, NAFOdown includes templates for producing SCR and STACFIS documents in Word format. Here we demonstrate how to use this package.

# Installation

The statistical computing program [R](https://www.r-project.org/) is required to use the NAFOdown package and we highly recommend using the [RStudio](https://www.rstudio.com/products/RStudio/) to manage an R projects. With RStudio installed a project can be started in an existing or new directory using the drop-down menus in Rstudio. After setting up a project, Rstudio will generate a .Rproj file which will automatically set the working directory to the project folder when this file is opened. This will simplify the process of telling R where to find the files needed for the report.

NAFOdown can be installed via GitHub using this code:

# install.packages("remotes")  
remotes::install\_github("nafc-assess/NAFOdown")

# Starting a document

A skeleton of a SCR or STACFIS document will be produced by running this code:

getwd() # FYI - this is where files and folders of the skeleton will be saved  
NAFOdown::draft(report\_type = "SCR")

The type of skeleton produced can be changed by changing the report\_type argument from "SCR" to "STACFIS". The components of the skeleton are outlined below. Note that all of the files included in the skeleton are text based and can be opened and edited using RStudio.

## Components of the skeleton

### index.Rmd

This is a key file that contains all the meta information that defines much of the formatting of the document as well as the information for the beginning of the document. Several of these items will need to be edited, such as the author, title, etc.

### 01\_body.Rmd, 02\_references.Rmd, etc.

These are the .Rmd files for each section of the report. These files will hold all the markdown text and R code needed to produce the document. Note that the SCR template includes a series of .Rmd files (01\_body.Rmd, 02\_references.Rmd, and 03\_appendix.Rmd), as these reports can be extensive, while the STACFIS template is simpler and only includes a body.Rmd file.

### \_bookdown.yml

This file outlines the .Rmd files to be combined by bookdown. This is also where some options are modified such as the prefix for table and figure captions (e.g. Figure instead of Fig.). Depending on the use case, specific components may not be needed and these files can be excluded from the .Rmd file list contained within this file (e.g. appendix.Rmd).

### csl

This folder holds the style files for bibliographies, specified using a citation style language file (.csl). Currently, SCR documents will follow the Harvard citation format. This folder is not included in the STACFIS template as references are rarely used.

### bib

BibTeX references are to be included in the refs.bib file inside this folder. BibTeX entries can be exported from most journal web-pages, Google scholar and from reference managers such as [Zotero](https://www.zotero.org/). Copy this text into the refs.bib file and note the tag used for each entry as this is what will be used to reference specific papers. Also consider using the Rstudio [citr](https://github.com/crsh/citr) add-in to simplify the insertion of citations.

# Writing

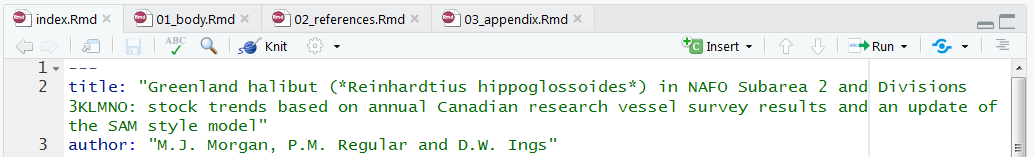
NAFOdown is build upon the [bookdown](https://bookdown.org/yihui/bookdown/) package which is build upon [R markdown](http://rmarkdown.rstudio.com). In short, R markdown provides a reproducible authoring framework for scientist as a single file can be used to embed chunks of code from R, and other languages, into a Markdown document. Markdown is a simple plain text formatting syntax that can be converted to a wide range of outputs, such as HTML. The format is easy-to-read and the fact that it detaches writing from formatting allows authors to focus on writing. Pairing R with Markdown creates a very flexible and efficient platform for producing a wide range of documents. See the bookdown book on bookdown for more details on its capabilities (<https://bookdown.org/yihui/bookdown/>).

# Meeting-to-meeting writing

As noted above, several of the files need to be populated with content. In most cases, the body.Rmd file will hold most of the content and take the most time to draft. Moreover, these documents may evolve over time. Managing a series of drafts can be a cumbersome task and, as such, we recommend that versions are tracked using a version control system such as Git. Rstudio integrates nicely with Git and the hosting service GitHub, and thereby provides a platform for syncing versions with an online repository (i.e. this acts as a back-up and facilitates remote collaboration). For a gentle guide to getting starting with using Git and GitHub with R and RStudio, see <http://happygitwithr.com/>.

# Rendering

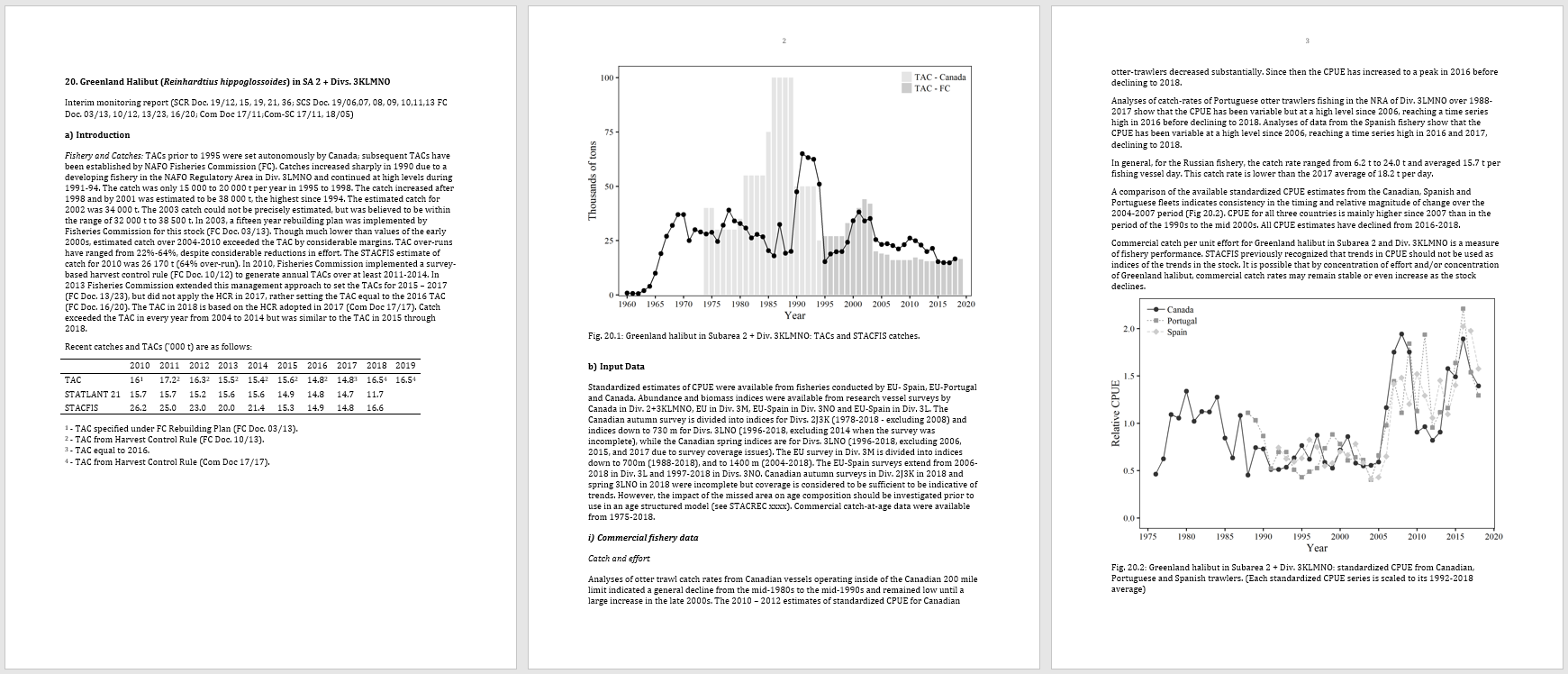
To render a NAFO report in Word format, open index.Rmd in Rstudio and click the “knit” button:



Documents can also be rendered in the R console, assuming the working directory is set to the directory holding the index.Rmd file, using bookdown’s render function:

bookdown::render\_book("index.Rmd")

The rendered Word file will be saved in the \_book/ directory. When all files are populated with the necessary content, the output will look similar to past documents:



however, these documents will be much easier to update following the addition of new data as all tables and figures should automatically update.