Package: TAMMsupport (via r-universe)

October 7, 2026

Title Streamline working with Terminal Area Management Modules

Version 1.1.0

Description A convenient tool for interfacing with Terminal Area Manamagement Modules (TAMMs) in R environments.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports cli, dplyr, glue, janitor, kableExtra, lifecycle, openxlsx, openxlsx2, purrr, quarto, readxl, rlang, stats, stringr, tibble, tidyr, tidyxl, utils, visNetwork, xldiff (>= 0.2.0)

Remotes cbedwards-dfw/xldiff

Depends R (>= 2.10)

LazyData true

URL https://framverse.r-universe.dev/TAMMsupport, https://github.com/cbedwards-dfw/TAMMsupport, https://cbedwards-dfw.github.io/TAMMsupport/

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev

Repository https://framverse.r-universe.dev

Date/Publication 2026-03-05 19:00:55 UTC

RemoteUrl https://github.com/cbedwards-dfw/TAMMsupport

RemoteRef HEAD

RemoteSha 680e61796e08607c4095db342460f8ba4172fe5b

Contents

address_parser . . . 3
as_numeric_safe . . . 3
chunk_formater_percenter . . . 4
chunk_formater_rounder . . . 4
chunk_read_2A_Cmrkd . . . 5
chunk_read_2A_CUandM . . . 5
chunk_read_2A_CUandM_N . . . 6
chunk_read_2A_CUnmrkd . . . 6
chunk_read_finisher . . . 7
clean_limiting_stock . . . 7
compare_chinook_inputs . . . 8
filter_tamm_wa . . . 9
find_color_matches . . . 10
fishery_renamer . . . 10
format_key_tamm_sheets_chin . . . 11
format_key_tamm_sheets_coho . . . 12
format_overview . . . 12
formula_formater . . . 13
fun_percenter . . . 14
fun_rounder . . . 14
kable_overview . . . 15
make_tracer_edges . . . 15
make_tracer_network . . . 16
make_tracer_nodes . . . 17
process_2ac_aeqs . . . 17
process_2ac_ers . . . 18
quick_er_format . . . 18
range_splitter . . . 19
read_2a_sheets . . . 19
read_2aCmrkd . . . 20
read_2aCUandM . . . 20
read_2aCUandM_N . . . 21
read_2aCUnmrkd . . . 21
read_jdf . . . 22
read_jdf_unmarked . . . 22
read_key_tamm_sheets_chin . . . 23
read_key_tamm_sheets_coho . . . 23
read_limiting_stock . . . 24
read_management_chin . . . 24
read_overview . . . 25
read_overview_complete . . . 26
read_tnt_allocation_chin . . . 26
reformat_management . . . 27
safe_convert_numeric . . . 28
tamm_compare . . . 28
tamm_compare3 . . . 29

tamm_diff 30
tamm_format_sheets 31
tamm_report 32
trace_formula 34
validate_data_frame 35

Index **37**

`address_parser` *Parse excel address including sheet*

Description

Helper function to parse full excel address into sheet, row, column. Removes ' from sheet names. If given a cell range, returns only the address of the first cell.

Usage

```
address_parser(address)
```

Arguments

`address`	Character string of excel address, including sheet.

Value

list, with `$sheet` giving sheet name, `$rows` givng the row number, `$cols` giving the column number.

Examples

```
address_parser("SPS!AS3")
address_parser("'Input Page'!$B$30")
```

`as_numeric_safe` *Safely convert character numerics into numeric*

Description

Function to translate columns that "should" be numeric into numeric but doesn't do so if that would mean converting a true character string to NA.

Usage

```
as_numeric_safe(x)
```

Arguments

x	Character vector

Value

Either numeric vector of x or (if that is lossy) character returns input.

`chunk_formater_percenter` *Format chunks of dataframe to present as %s, round digits*

Description

Format chunks of dataframe to present as %s, round digits

Usage

```
chunk_formater_percenter(df, block_ranges, percent_digits = 1)
```

Arguments

`df`	dataframe of sheet to apply formatting to.
`block_ranges`	vector of characters specifying blocks of cells (in excel nomenclature) to format as %s
`percent_digits`	Decimal place to round to in percent

Value

Formatted version of `df`

`chunk_formater_rounder` *Format chunks of dataframe to round digits*

Description

Format chunks of dataframe to round digits

Usage

```
chunk_formater_rounder(df, block_ranges, digits = 1)
```

Arguments

df	dataframe of sheet to apply formatting to.
block_ranges	vector of characters specifying blocks of cells (in excel nomenclature) to format as %s
digits	Decimal place to round to.

Value

Formatted version of df

---

chunk_read_2A_Cmrkd *Read in chunks of the 2A_Cmrkd sheet*

Description

Read in chunks of the 2A_Cmrkd sheet

Usage

```
chunk_read_2A_Cmrkd(full_sheet, start_col, end_col, table_name)
```

Arguments

full_sheet	entire raw spreadsheet
start_col	Letter of the column the chunk starts on
end_col	Letter of the column the chunk ends on
table_name	name of table

---

chunk_read_2A_CUandM *Read chunk for table 2A_CUandM*

Description

Read chunk for table 2A_CUandM

Usage

```
chunk_read_2A_CUandM(full_sheet, start_col, end_col, table_name)
```

Arguments

full_sheet	entire raw spreadsheet
start_col	Letter of the column the chunk starts on
end_col	Letter of the column the chunk ends on
table_name	name of table

`chunk_read_2A_CUandM_N` *Read in individual chunks for sheet 2A_Cu&M_N*

Description

Read in individual chunks for sheet 2A_Cu&M_N

Usage

```
chunk_read_2A_CUandM_N(full_sheet, start_col, end_col, table_name)
```

Arguments

`full_sheet`	entire raw spreadsheet
`start_col`	Letter of the column the chunk starts on
`end_col`	Letter of the column the chunk ends on
`table_name`	name of table

`chunk_read_2A_CUnmrkd` *Reads chunk of 2A_CUnmrkd sheet*

Description

Reads chunk of 2A_CUnmrkd sheet

Usage

```
chunk_read_2A_CUnmrkd(full_sheet, start_col, end_col, table_name)
```

Arguments

`full_sheet`	entire raw spreadsheet
`start_col`	Letter of the column the chunk starts on
`end_col`	Letter of the column the chunk ends on
`table_name`	name of table

chunk_read_finisher *Finishes a chunk after initial reading*

Description

Handles some of the repetitive tasks of processing individual chunks; this part is generalized and does not need to be different for each sheet. Lots of fiddliness in here because TAMM formatting is terrible

Usage

```
chunk_read_finisher(
  sheet_name,
  table_name,
  raw,
  raw_titles,
  data_er,
  do_er = TRUE
)
```

Arguments

sheet_name	Name of sheet
table_name	Name of table
raw	.
raw_titles	.
data_er	.
do_er	Should ER be done? If FALSE, skips ER (useful for processing sections with no ER.)

Value

A list with $aeq and $er, each a dataframe

clean_limiting_stock *Generates clean read of TAMM limiting stock sheet*

Description

Effectively a wrapper function for read_limiting_stock with some formatting added in. Filters to unmarked naturals, just present ER values.

Usage

```
clean_limiting_stock(file)
```

Arguments

file	Name (and path) for TAMM files

Value

dataframe

---

compare_chinook_inputs

Compare differences in input sheets of two TAMM files

Description

Summarizes the differences in input sheets and returns as a table. Wrapper for `xldiff::excel_diff_table()`

Usage

```
compare_chinook_inputs(
  file_1,
  file_2,
  proportional_threshold = 0.001,
  absolute_threshold = NULL,
  digits_show = 6,
  trim_cols = TRUE,
  diff_only = TRUE
)
```

Arguments

file_1	Filename of first TAMM file
file_2	Filename of second TAMM file
proportional_threshold	Sets a threshold of proportional change below which differences should be ignored. For example, a value of 0.1 means any changes less than 10% will not be flagged as having changed. `proportional_threshold` will override this value and behavior if it is provided. Numeric, defaults to 0.001 (0.1% change).
absolute_threshold	Optional. Sets a threshold of absolute change below which differences should be ignored. For example, a value of 0.1 means any changes less than 0.1 will not be flagged as having changed. If provided, will override `proportional_threshold`. Numeric, defaults to NULL.
digits_show	When there is a change in number values, how many digits should be shown in `## ---> ##`? Numeric, defaults to 6. Recommend not making this so small that flagged changes don't get printed (e.g., if this is 2 and `proportional_threshold` is 0.001, 0.1% changes will get flagged, but only the first two digits will get shown).

trim_cols	Should columns that contain no differences be cut from the results? Defaults to TRUE.
diff_only	Should entries that do not contain differences be replaced by "" to help spot differences? Defaults to TRUE.

Value

Tibble of differences in the input files; only rows with differences are included. row_name variable gives the row name in the TAMM file, and column names match excel column names.

---

filter_tamm_wa *Filters a dataframe to Washington State fisheries.*

Description

Adapted from framrsquared::filter_wa(), but (a) only for Chinook (at present), and (b) uses TAMM fishery ids, so includes ids 72 and 73. Includes code for COHO based on FRAM ids. filter_tamm_wa() uses attributes to specify chinook or coho. To directly filter for chinook or coho, use filter_tamm_wa_chin() or filter_tamm_wa_coho().

Usage

```
filter_tamm_wa(.data)

filter_tamm_wa_chin(.data)

filter_tamm_wa_coho(.data)
```

Arguments

.data	Dataframe generated within this package

Details

a fishery_id column name.

Examples

```
## Not run:
fram_dataframe |> filter_wa()

## End(Not run)
```

find_color_matches *Identify all cells in Excel sheet or sheets with matching cell colors*

Description

Designed for tracking the status of TAMM inputs, which are typically color coded. Searches through specified sheet or sheets, looking for cells whose colors match the `target_cell` argument. Returns a condensed list of those entries.

Usage

```
find_color_matches(file, target_cell, sheets = "Input Page")
```

Arguments

`file`	Character string of excel file name, including path if relevant
`target_cell`	Character string of excel address of cell with the cell shading of interest (e.g., "B4")
`sheets`	character string or character vector of sheets to search through

Value

tibble summarizing entries with the color of interest, including sheet, row and column ids, and row name (entry of first column in sheet)

Examples

```
## Not run:
file = "chin tamms/Chin2023_Final.xlsx"
find_color_matches(file, "B20")

## End(Not run)
```

fishery_renamer *relabel inconsistent fishery names in FRAM/TAMM*

Description

With argument `sep = TRUE`, will return data frame with separate columns for the area and the the "class" (Net, Troll, Sport)

Usage

```
fishery_renamer(x, sep = FALSE)
```

Arguments

x	character vector of fishery names from TAMM or FRAM
sep	Should we return a dataframe with separate column for class of fishing? Defaults to FALSE

Value

#vector with cleaned fishery names OR dataframe with cleaned name (`$full`), area without fishery type (`$area`) and type of fishery (`$class`).

---

format_key_tamm_sheets_chin

[Superseded] *Modify list of Chinook TAMM spreadsheet dataframes to facilitate comparison.*

Description

Function is no longer needed for tamm_diff now that openxlsx

Usage

```
format_key_tamm_sheets_chin(
  dat,
  percent_digits = 1,
  numeric.digits = 1,
  numeric.digits.small = 4
)
```

Arguments

dat	list of dataframes corresponding to the overview, limiting stock, and inputs tabs. Must be named `$overview`, `$limiting`, and `$input`
percent_digits	Optional, number of decimal places to round percentages to before comparison. Defaults to 1.
numeric.digits	Optional, number of decimal places to round non-percentage numerics to before comparison. Applied to numbers that are expected to have natural units of whole numbers (e.g. numbers of fish). Defaults to 1.
numeric.digits.small	Optional, number of decimal places to round non-percentage numerics to before comparison. Applied to numbers that are expected to be small (e.g. rates, proportions) Defaults to 4.

Value

list of dataframes with same structure as `dat`, contents modified.

format_key_tamm_sheets_coho

Modify list of Coho TAMM spreadsheet dataframes to facilitate comparison.

Description

Modify list of Coho TAMM spreadsheet dataframes to facilitate comparison.

Usage

```
format_key_tamm_sheets_coho(
  dat,
  percent_digits = 2,
  numeric.digits = 1,
  numeric.digits.small = 4
)
```

Arguments

dat	list of dataframes corresponding to the overview, limiting stock, and inputs tabs. Must be named $overview, $limiting, and $input
percent_digits	Optional, number of decimal places to round percentages to before comparison. Defaults to 1.
numeric.digits	Optional, number of decimal places to round non-percentage numerics to before comparison. Applied to numbers that are expected to have natural units of whole numbers (e.g. numbers of fish). Defaults to 1.
numeric.digits.small	Optional, number of decimal places to round non-percentage numerics to before comparison. Applied to numbers that are expected to be small (e.g. rates, proportions) Defaults to 4.

Value

list of dataframes with same structure as dat, contents modified.

format_overview

Format TAMM overview for easy printing

Description

Format TAMM overview for easy printing

Usage

```
format_overview(dat_overview)
```

Arguments

dat_overview	dataframe of TAMM overview page, as produced by read_overview()

Value

list with the formatted data, a vector of indent information, and a vector of bolding information.

---

formula_formater *Add spaces to excel formulas for readability*

Description

Helper function to add spaces to excel formulas to improve readability. Can also add newlines based on a character or regular expression

Usage

```
formula_formater(x, newline_breakpoint = NULL)
```

Arguments

x	Character string containing an excel formula
newline_breakpoint	Pattern to identify locations for newlines. Could be a single character (e.g. "/") or a more complex regular expression (e.g., "[)][*-+/]" to add a newline after a combination of an end parathesis and an operator).

Value

character string

Examples

```
formula_formater("(SUM(A1:A4)*5/11)/(B3-B4)")
formula_formater("(SUM(A1:A4)*5/11)/(B3-B4)", newline_breakpoint = "/")
formula_formater("(SUM(A1:A4)*5/11)/(B3-B4)", newline_breakpoint = "[*-+/]")
```

fun_percenter *Format vector of mixed characters to present numbers as percents*

Description

Intended for internal use within formatting functions. Note that percents in excel are read in as proportions in R – this makes them percents to make saved files more readable.

Usage

```
fun_percenter(x, percent_digits)
```

Arguments

x	Character vector, presumably containing some entries that are numbers in character form
percent_digits	Number of digits to round to after converting to percents

Value

character vector with individual entries converted to percentages and rounded as appropriate.

fun_rounder *Format vector of mixed characters to round numbers to specified digits*

Description

Intended for internal use within formatting functions.

Usage

```
fun_rounder(x, digits)
```

Arguments

x	Character vector, presumably containing some entries that are numbers in character form
digits	Number of digits to round numeric items to

kable_overview *Print table of overview using kable and kableExtra*

Description

Uses output of format_overview()

Usage

```
kable_overview(dat_overview, ind_indent, col_bold)
```

Arguments

dat_overview	formatted data, first item of format_overview() output
ind_indent	Vector of indices that need indenting to match TAMM overview formatting. Second item of format_overview() output
col_bold	vector of names for bolding to match TAMM overview formatting. Third item of format_overview() output.

Value

html table

make_tracer_edges *Calculate and format edges from traced objects*

Description

Helper function to find all references in list created within tracer_formula, and create a simple dataframe representing all the "edges" of the traced network (that is, all the connections between cells). Gets called in `trace_formula()` to generate the `$references` component of the output.

Usage

```
make_tracer_edges(address_list)
```

Arguments

address_list	The `$raw.tracing` component of the results of a `trace_formula()` call.

Value

tibble. `$from` is the index of the cell that is referenced, `$to` is hte index of the cell that is doing the referencing `$from_name` and `$to_name` are the excel addresses of the same.

`make_tracer_network` *Visualize excel formula tracing*

Description

[Experimental]

Usage

```
make_tracer_network(tracer_list, save_path = NULL, newline_breakpoint = NULL)
```

Arguments

`tracer_list`	Output of a `trace_formula()` call.
`save_path`	Optional, defaults to NULL. Provide a file path + name to save an html of the visualization. Make sure file name ends in `.html`.
`newline_breakpoint`	Optional character string used to add breakpoints to formulas to improve readability. Can be simple character or regular expression; See `formula_formater` for details and examples.

Details

Render the dependency network created in `trace_formula()`; optionally save this to an interactive .html.

Value

interactive visNetwork object

Examples

```
## Not run:
trace_network = trace_formula(path = "NOF material/NOF 2024/NOF 2/Chin1624.xlsx",
cell_start = "SPSmrkd!AS20")
make_tracer_network(trace_network)

## End(Not run)
```

make_tracer_nodes *List cells and contents from traced objects*

Description

Helper function to parse list created within trace_formula into a tibble with each cell represented by a row. Called in `trace_formula()` to generate the `$cells` component of the output.

Usage

```
make_tracer_nodes(address_list)
```

Arguments

`address_list`	The `$raw.tracing` component of the results of a `trace_formula()` call.

Value

tibble summarizing each cell in the tracing. See `trace_formula()` for details.

process_2ac_aeqs **[Experimental]** *Read and combine aeq sections form tables 2a-2c*

Description

[Experimental] Read and combine aeq sections form tables 2a-2c

Usage

```
process_2ac_aeqs(tab_2a, tab_2b, tab_2c)
```

Arguments

`tab_2a`	the "Table 2a" section of one of the 2ac sheets, read in with a `chunk_reader_2A_*()` function
`tab_2b`	same, but 2b
`tab_2c`	same, but 2c

Value

dataframe of AEQ values for each stock in longform

`process_2ac_ers` **[Experimental]** *function to process + combine the er sections from tables 2a-2c, used below*

Description

[Experimental] function to process + combine the er sections from tables 2a-2c, used below

Usage

```
process_2ac_ers(tab_2a, tab_2b, tab_2c)
```

Arguments

`tab_2a`	the "Table 2a" section of one of the 2ac sheets, read in with a `chunk_reader_2A_*()` function
`tab_2b`	same, but 2b
`tab_2c`	same, but 2c

Value

dataframe of ers in longform

`quick_er_format` *Quick helper function to format vectors that contain text and ERs*

Description

Quick helper function to format vectors that contain text and ERs

Usage

```
quick_er_format(x)
```

Arguments

`x`	character vector

range_splitter *Translate excel-style cell range to vector of cell addresses.*

Description

Primarily intended as a helper function for `trace_formula()`, but could be useful for other excel tasks. Drops `$` and `'` symbols from addresses to simplify parsing.

Usage

```
range_splitter(address)
```

Arguments

`address`	Character string of excel cell range, as in "B10:C15" or "'Input Sheet'!AS30:AS40"

Value

vector of individual excel-style cell addresses within that range

Examples

```
range_splitter("B10:C15")
range_splitter("'Input Sheet'!$AS30:$AS40")
```

read_2a_sheets *Read and combine all 2a sheets from a TAMM file*

Description

Gathers and cleans the AEQ and ER information for tamm sheets `2aCmrkd`, `2aCUnmkrd`, `2aCUandM`, and `2aCUandM_N`.

Usage

```
read_2a_sheets(tamm_filepath, stock_cleanup = TRUE)
```

Arguments

`tamm_filepath`	TAMM filepath
`stock_cleanup`	Clean up stock name capitalization? Logical, defaults to TRUE.

Value

list, with $aeq containing the primary information (AEQs of fishery impacts on each stock) and $er containing the ER information at the bottom of the spreadsheets.

Examples

```
## Not run:
library(here)
filepath <- here("tamms/Chin2025.xlsx")
res <- read_2a_sheets(filepath)
jdf <- read_jdf(filepath)

## End(Not run)
```

`read_2aCmrkd` *Reads sheet of 2A_Cmrkd*

Description

Reads sheet of 2A_Cmrkd

Usage

```
read_2aCmrkd(tamm_filepath, stock_cleanup = TRUE)
```

Arguments

`tamm_filepath`	path to file
`stock_cleanup`	Do cleanup of capitalization?

Value

a list, with $aeq (the main parts of each table) and $er (the very bottom section)

`read_2aCUandM` *Read and process all parts of sheet 2A CU and M*

Description

Read and process all parts of sheet 2A CU and M

Usage

```
read_2aCUandM(tamm_filepath, stock_cleanup = TRUE)
```

Arguments

`tamm_filepath`	Tamm filepath
`stock_cleanup`	Clean up weird capitalization? Logical, defaults to TRUE

Value

a list, with $aeq (the main parts of the table) and $er (the very bottom section) @return

`read_2aCUandM_N` *Read and combine all chunks for sheet 2A CU and M_N*

Description

Read and combine all chunks for sheet 2A CU and M_N

Usage

```
read_2aCUandM_N(tamm_filepath, stock_cleanup = TRUE)
```

Arguments

`tamm_filepath`	Tamm filepath
`stock_cleanup`	Clean up stock name capitalization? Logical, defaults to TRUE

`read_2aCUnmrkd` *read all info for 2aCUnmrkd and process it.*

Description

read all info for 2aCUnmrkd and process it.

Usage

```
read_2aCUnmrkd(tamm_filepath, stock_cleanup = TRUE)
```

Arguments

`tamm_filepath`	Tamm filepath
`stock_cleanup`	Logical, defaults to TRUE. Convert weird capitalization to lowercase?

Value

a list, with $aeq (the main parts of the table) and $er (the very bottom section)

read_jdf — *Read in AEQ and ER information from the JDF sheet*

Description

Read in AEQ and ER information from the JDF sheet

Usage

```
read_jdf(tamm_filepath, stock_cleanup = TRUE)
```

Arguments

tamm_filepath	TAMM filepath
stock_cleanup	Clean up stock name capitalization? Logical, defaults to TRUE.

Value

List, with $aeq containing the combined AEQ information that is spread across multiple chunks in the TAMM sheet. List also contains $er = NULL, to provide parallel outputs to `read_2a_sheets`, which *does* have a non-null $er item in the output list.

Examples

```
## Not run:
library(here)
filepath <- here("tamms/Chin2025.xlsx")
jdf <- read_jdf(filepath)

## End(Not run)
```

read_jdf_unmarked — *Read in AEQ and ER information from the JDF Unmarked sheet*

Description

Read in AEQ and ER information from the JDF Unmarked sheet

Usage

```
read_jdf_unmarked(tamm_filepath, stock_cleanup = TRUE)
```

Arguments

tamm_filepath	TAMM filepath
stock_cleanup	Clean up stock name capitalization? Logical, defaults to TRUE.

Value

List, with $aeq containing the combined AEQ information that is spread across multiple chunks in the TAMM sheet. List also contains $er = NULL, to provide parallel outputs to `read_2a_sheets`, which *does* have a non-null $er item in the output list.

Examples

```
## Not run:
library(here)
filepath <- here("tamms/Chin2025.xlsx")
jdf <- read_jdf_unmarked(filepath)

## End(Not run)
```

---

read_key_tamm_sheets_chin

Read Chinook TAMM files to extract the key sheets

Description

Read Chinook TAMM files to extract the key sheets

Usage

```
read_key_tamm_sheets_chin(file)
```

Arguments

`file`	#Tamm file name

Value

List of dataframes: `$overview`, `$limiting`, and `$input`

---

read_key_tamm_sheets_coho

Read Chinook TAMM files to extract the key sheets

Description

Read Chinook TAMM files to extract the key sheets

Usage

```
read_key_tamm_sheets_coho(file)
```

Arguments

file	#Tamm file name

Value

List of dataframes: $two, $tami, and $wacoast

---

read_limiting_stock *Read TAMM limiting stock complete tab*

Description

Reads in the table, adds fishery type information.

Usage

```
read_limiting_stock(file, longform = FALSE)
```

Arguments

file	Tamm name (and path)
longform	Should results be in long form (good for R stuff) (TRUE) or replicate the structure of the TAMM sheet (FALSE). Logical, defaults to FALSE.

Value

data frame summarizing the TAMM limiting stock tab. The "block" of the limiting tab is identified with column "stock_type". "first_section" is the first section in the TAMM; this section is primarily natural fish but includes some hatchery fish if they were included in management objectives. "ALL_H" is all hatchery fish (in TAMM, block starts on row 81), "UM_H" is unmarked hatchery (in TAMM, starts on row 160), "UM_N" is unmarked naturals (in TAMM, starts row 239), "AD_H" is marked hatchery (in TAMM, starts row 318), and "AD_N" is marked naturals (unless programs begin marking natural fish, this should always be all 0s; in TAMM starts row 397).

---

read_management_chin *Extract and format management criterion from Chinook TAMM*

Description

Extract and format management criterion from Chinook TAMM

Usage

```
read_management_chin(file)
```

Arguments

file	filename, including filepath if appropriate

Value

Tibble, with $management_name giving the name of the management criteria as specified in the TAMM, $management_criteria giving a list of dataframes with the associated management criteria, formatted appropriately, and $notes giving a list of character vectors with notes (if any) to help interpret the criteria.

Examples

```
## Not run:
cur.management = read_management_chin("Code Inputs/Pre-Season/TAMMs/2013_Final W_BP7.1.xlsx")

## End(Not run)
```

---

read_overview *Read overview sheet from TAMM and return key stock*

Description

Read overview sheet from TAMM and return key stock

Usage

```
read_overview(path)
```

Arguments

path	filename (including path) to Chinook TAMM

Value

dataframe

`read_overview_complete` *Read overview sheet from TAMM and return all stock*

Description

The stock column of the overview tab uses R-unfriendly approaches to store season, overall stock name, and individual stocks in the `stock` column of the TAMM. `read_overview_complete` splits these into separate columns: `season`, `primary_stock`, and `stock`. For three stock (Green, Puyallup, Skokomish), the tamm presents model predictions across two rows with a single stock name due to merged cells. The `stock` column in the output preserves this information; the entry corresponding to the second row has the suffix "_row2".

Usage

```
read_overview_complete(path)
```

Arguments

`path`	filename (including path) to Chinook TAMM

Value

Dataframe containing all infomration from cells A2:H34 of the "ER_ESC_Overview_New" sheet.

`read_tnt_allocation_chin` **[Experimental]**

Description

Extract Treaty and Nontreaty numbers from "2A_CU&M_H+N" TAMM sheet. For Elwha and Dungeoness, separate allocations are calculated using column O of the JDF tab

Usage

```
read_tnt_allocation_chin(file)
```

Arguments

`file`	Character vector. Filename (including path) for chinook TAMM

Details

Currently provides dataframe with `stock.original` column identifying the stock based on their names in the TAMM, and `stock.clean` with more human-readable names (which are generally consistent with other stock names, like those used to label the management criterion).

Value

dataframe with the treaty and nontreaty mortalities at each stock.

`reformat_management` *Convenience function for formatting TAMM management criterion chunks*

Description

Internal tool to simplify converting TAMM sheet to stock management criterion lists. See `read_management_chin()` for example use.

Usage

```
reformat_management(
  df,
  cols.percent,
  rows.percent = NULL,
  rows.header = 2,
  notes = NULL
)
```

Arguments

`df`	dataframe trimmed from TAMM management page to just the relevant stock's criterion.
`cols.percent`	Column numbers for columns that should be converted to percents.
`rows.percent`	Character vector of row sub-stock names identifying rows (if any) that should be translated to percents. Defaults to NULL. Developed to accomodate "CERC (SUS)" in Snohomish.
`rows.header`	How many rows are used to make up the header? Defaults to 2, but sometimes header block in TAMM is 1 row.
`notes`	Optional. Character vector, each item of which is a separate note associated with this stock. Generally based on comments in excel doc, can also feed in "extra" rows from TAMM that contain caveats.

Value

Stock management criterion list, containing dataframe extracted from TAMM and notes hard-coded in based on TAMM cell comments.

safe_convert_numeric *Convert string of number to numeric*

Description

Helper function to check an atomic string and convert to numeric if it contains a legal number.

Usage

```
safe_convert_numeric(x)
```

Arguments

x	character string

Value

Input formatted as character string or numeric

Examples

```
safe_convert_numeric("AS20-BC2")
safe_convert_numeric("10.5")
```

tamm_compare *Generate summary comparison of any number of TAMM files*

Description

Intended use case is to compare low, mid, and high ocean option.

Usage

```
tamm_compare(
  tamm_names,
  tamm_path = getwd(),
  fisheries = NULL,
  clean = TRUE,
  overwrite = TRUE
)
```

Arguments

tamm_names	character vector of the three tamm files to compare (including .xlsx suffix).
tamm_path	Absolute path to directory containing the tamm file. here::here() can be useful in identifying appropriate path. Character atomic; defaults to current working directory.
fisheries	Optional, numeric vector of fishery IDs to filter to before plotting fishery-specific ER for each stock. Defaults to NULL (no filtering).
clean	Should the intermediate .qmd files used to make the report be deleted afterwards? Logical, defaults to TRUE. Set to FALSE to explore the .qmd files underlying the report.
overwrite	Should the intermediate .qmd files or the final report be overwritten if those files already exist?; Logical, defaults to TRUE.

Value

Nothing

Examples

```
## Not run:
tamm_path <- "C:/Users/edwc1477/Documents/WDFW FRAM team work/NOF material/NOF 2024/FRAM"
tamm_names <- c("Chin1024.xlsx", "Chin1124.xlsx", "Chin1224.xlsx")
tamm_compare3(tamm_names = tamm_names, tamm_path = tamm_path)

## End(Not run)
```

---

tamm_compare3 *Generate summary comparison of 3 TAMM files*

Description

Intended use case is to compare low, mid, and high ocean option.

Usage

```
tamm_compare3(tamm_names, tamm_path = getwd(), clean = TRUE, overwrite = TRUE)
```

Arguments

tamm_names	character vector of the three tamm files to compare (including .xlsx suffix).
tamm_path	Absolute path to directory containing the tamm file. here::here() can be useful in identifying appropriate path. Character atomic; defaults to current working directory.
clean	Should the intermediate .qmd files used to make the report be deleted afterwards? Logical, defaults to TRUE. Set to FALSE to explore the .qmd files underlying the report.
overwrite	Should the intermediate .qmd files or the final report be overwritten if those files already exist?; Logical, defaults to TRUE.

Value

Nothing

Examples

```
## Not run:
tamm_path <- "C:/Users/edwc1477/Documents/WDFW FRAM team work/NOF material/NOF 2024/FRAM"
tamm_names <- c("Chin1024.xlsx", "Chin1124.xlsx", "Chin1224.xlsx")
tamm_compare3(tamm_names = tamm_names, tamm_path = tamm_path)

## End(Not run)
```

`tamm_diff` *Compare key sheets of TAMM files*

Description

Compare key sheets of TAMM files

Usage

```
tamm_diff(
  file_1,
  file_2,
  results_name,
  proportional_threshold = 0.001,
  absolute_threshold = NULL,
  extra_width = NULL
)
```

Arguments

`file_1`	name of first TAMM file to compare. Include file path if file is not in working directory.
`file_2`	name of second TAMM file to compare. Include file path if file is not in working directory.
`results_name`	name of output sheets. Include file path if save location is not in working directory.
`proportional_threshold`	Sets a threshold of proportional change below which differences should be ignored. For example, a value of 0.1 means any changes less than 10% will not be flagged as having changed. `proportional_threshold` will override this value and behavior if it is provided. Numeric, defaults to 0.001 (0.1% change).
`absolute_threshold`	Optional. Sets a threshold of absolute change below which differences should be ignored. For example, a value of 0.1 means any changes less than 0.1 will not be flagged as having changed. If provided, will override `proportional_threshold`. Numeric, defaults to NULL.

extra_width	How much extra width should be added to columns that changed? Helpful to improve readability, since changed cells have longer entries. Numeric, defaults to 0.4.

Examples

```
## Not run:
tamm_diff(
  file_1 = here("FRAM/Chin1124.xlsx"),
  file_2 = here("NOF 2/Chin2524.xlsx"),
  results_name = here("Chin 1124 vs Chin 2524.xlsx")
)

## End(Not run)
```

---

tamm_format_sheets *Apply final formatting to diff'd TAMM sheets*

Description

Functions that add formatting to broadly replicate the formatting of the TAMM sheets. Colored foregrounds have been toned down to help change highlighting pop, and some of the complicated or superfluous formatting has been skipped.

Usage

```
tamm_format_overview(wb, diff.sheet, tabname = "overview")

tamm_format_limiting(wb, tabname = "limiting")

tamm_format_input(wb, tabname = "input", wrap.text = FALSE)

tamm_format_wacoast(wb, tabname = "wacoast")
```

Arguments

wb	openxlsx workbook object containing $overview, limiting and $input sheets with the contents of diffing two TAMM files.
diff.sheet	For tamm_format_overview, the output of xldiff::sheet_comp of the overview tab. Used to programmatically bold the appropriate ERs based on the management objective.
tabname	Name of sheet in the wb to be modified. Defaults to correct value.
wrap.text	Should specific cells with long content use text wrapping? Defaults to FALSE.

`tamm_report` *Generate report of figures from TAMM file*

Description

Generate report of figures from TAMM file

Usage

```
tamm_report(
  tamm_name,
  tamm_path = getwd(),
  clean = TRUE,
  overwrite = TRUE,
  additional_children = NULL,
  additional_support_files = NULL
)
```

Arguments

`tamm_name`	Name of tamm file (including .xlsx suffix). Character atomic
`tamm_path`	Absolute path to directory containing the tamm file. `here::here()` can be useful in identifying appropriate path. Character atomic; defaults to current working directory.
`clean`	Should the intermediate .qmd files used to make the report be deleted afterwards? Logical, defaults to `TRUE`. Set to `FALSE` to explore the .qmd files underlying the report.
`overwrite`	Should the intermediate .qmd files or the final report be overwritten if those files already exist?; Logical, defaults to `TRUE`.
`additional_children`	Optional argument with filepath(s) to additional quarto child documents (see `Details`). Defaults to `NULL`.
`additional_support_files`	Optional argument with filepath(s) to additional files needed by additional quarto documents. Files will be copied into the same directory as the TAMM for easy reading/use by `additional_children` quarto files, and then deleted after the report is generated

Details

This function generates a summary report of a single chinook TAMM (Terminal Area Management Module) file. The common use case is to call this function for a tamm, and then view the resulting html report (which will be generated in the same folder as the TAMM). However, this function also provides substantial flexibility in the form of optional user-created child quarto documents, which requires some additional context to develop and use.

tamm_report is implemented using a parameterized quarto report and a child quarto document, which are included in the package. The resulting report includes some broadly useful content – currently, a replication of the overview TAMM sheet and bar charts showing the breakdown of exploitation rates by fishery for each stock listed in the limiting stock complete TAMM sheet. However, individuals or organizations may consistently want additional, specific visualizations based on their own needs. For example, individual tribes may want to visualize impacts on a single stock, but use AEQ instead of ER rates. The tamm_report function is designed to integrate these extra components with the additional_children argument.

The basic principle is that the individual or group can write a reusable "child" .qmd file, see this explanation using Rmarkdown), which will effectively be inserted into the main report. As an example, we might create a simple additional child .qmd file called extra-tamm-child.qmd to include a small table summarizing a few key aspects of the Nooksack Earlies stock. The contents of that file are provided in the example below.

When we call tamm_report(), we can include as an argument additional_children = with the file name (including file path) for our new file. In this way we can easily include this extra content any time we create a tamm report. If desired, multiple child files can be created, and additional_children can take a character vector with each child file name.

When writing child documents, formatting and chunks work akin to copy-pasting a section of a quarto file out of the main file. (To easily access the main quarto file, run tamm_report() with clean = FALSE, which will leave tamm-visualizer.qmd and tamm-visualizer-child.qmd files in the folder with the TAMM file). The child document will have access to key objects

- dat.all: the output of read_limiting_stock with longform = TRUE called on the TAMM;
- dat_overview: the output of of read_overview called on the TAMM;
- and dat: the specific filtered and formatted version of the limiting stock data generated in the limiting stock tab chunk of tamm-visualizer.qmd and used in generating the unmarked natural stock exploitation rate figures. When developing a child document, it may be useful to create in the working environment versions of dat.all and dat_overview from an example TAMM to facilitate writing appropriate filters, plot-making code, etc. The primary quarto report loads in the tidyverse, framrsquared, kableExtra, and TAMMsupport packages, but you can include additional packages for the child documents as needed by adding library(packagename calls just like in any other context.

If the additional_children files need to read contents of additional files (or call children .qmd files of their own to streamline looping over stocks or fisheries), the file paths to these supplemental files can be provided in the additional.support files argument. These files will be copied into the same folder as the .qmd files, so the children quarto files can be written to access the supplemental files without dealing with file paths.

For questions/support in developing child documents to extend the report, contact Collin Edwards at WDFW.

Value

Nothing

Examples

```
## Not run:
```

```
##### CONTENTS FOR extra-tamm-child.qmd #####----------------------------
## Test section

Hypothetical world in which we want a table summarizing the impacts of three
fisheries on the "Nooksack Earlies" stock.

```{r}
kable(dat |> filter(stock == "Nooksack Earlies") |> filter(fishery_id %in% 16:18))
```
END CONTENTS FOR extra-tamm-child.qmd #####--------------------------

tamm_path = "C:/Users/edwc1477/Documents/WDFW FRAM team work/NOF material/NOF 2024/NOF 2"
tamm_name = "Chin2124.xslx"
tamm_report(tamm_name = "Chin2124.xlsx", tamm_path = tamm_path)
Now with our extra content, presuming `extra-tamm-child.qmd` is saved in our documents folder
tamm_report(tamm_name = "Chin2124.xlsx", tamm_path = tamm_path,
  additional_children = "C:/Users/JohnDoe/Documents/extra-tamm-child.qmd")

End(Not run)
```

`trace_formula` *Trace the calculations of a cell recursively through all referenced cells*

Description

[Experimental]

Usage

```
trace_formula(
  path,
  cell_start,
  max_it = 5000,
  verbose = TRUE,
  split_ranges = FALSE
)
```

Arguments

`path`	Filepath of an excel file
`cell_start`	Character string of intiial cell in excel format. MUST include sheet name (e.g. "SPS!AS20", not "AS20")
`max_it`	Maximum iterations; used as a failsafe to ensure function eventually in case of circular error. Defaults to 5000; increase if actual dependency network is likely to have more than 5000 nodes.
`verbose`	Print cell addresses to console during tracing? Logical, defaults to `FALSE`.

split_ranges When encountering a reference that includes a cell range, trace backwards for all cells (TRUE) or just the first cell in the range (FALSE)? Logical, defaults to FALSE. This option was added because sometimes formulas include sums across large ranges, ballooning the size of the resulting network. For building understanding, it is sometimes sufficient to trace only a representative from each referenced range, leading to smaller and simpler plots.

Value

"trace object" – list defining the dependency network.

$cells tibble summarizing each of the cells in the dependency network, starting with the cell_start. Within this, $id is an index; $label is the cell address; $formula is the formula in that cell if there is a formula, otherwise it is the contents of the cell, $contents is the non-formula cell contents (i.e., if a formula is present, $contents will be the results of the formula); $is.formula is a logical identifying if this cell contains a formula, or is purely a numeric / string / etc contents; $addresses.referenced is a character vector of all excel addresses in $formula, and sheet is the sheet associated with the current cell.

$references is a tibble summarizing each of the edges of the dependency network – that is, all of the references in one cell to another. $from is the index of the cell that is referenced; $to is the index of the cell that is doing the referencing; $from_name and $to_name are the excel addresses of the same.

$raw.tracing list created during tracing that forms the backbone of trace_formula. Intended for internal use; $cells and $references contains the important results here.

Examples

```
## Not run:
trace_network = trace_formula(path = "NOF material/NOF 2024/NOF 2/Chin1624.xlsx",
 cell_start = "SPSmrkd!AS20")

## End(Not run)
```

---

validate_data_frame *Confirm object is dataframe*

Description

Confirm object is dataframe

Usage

```
validate_data_frame(
  x,
  ...,
  arg = rlang::caller_arg(x),
  call = rlang::caller_env()
)
```

Arguments

x	Object to check
...	additional arguments
arg	name of argument in calling function, identified programmatically using rlang
call	name of calling function, identified programmatically using rlang

Index

address_parser, 3
as_numeric_safe, 3

chunk_formater_percenter, 4
chunk_formater_rounder, 4
chunk_read_2A_Cmrkd, 5
chunk_read_2A_CUandM, 5
chunk_read_2A_CUandM_N, 6
chunk_read_2A_CUnmrkd, 6
chunk_read_finisher, 7
clean_limiting_stock, 7
compare_chinook_inputs, 8

filter_tamm_wa, 9
filter_tamm_wa_chin *(*filter_tamm_wa*)*, 9
filter_tamm_wa_coho *(*filter_tamm_wa*)*, 9
find_color_matches, 10
fishery_renamer, 10
format_key_tamm_sheets_chin, 11
format_key_tamm_sheets_coho, 12
format_overview, 12
formula_formater, 13
fun_percenter, 14
fun_rounder, 14

kable_overview, 15

make_tracer_edges, 15
make_tracer_network, 16
make_tracer_nodes, 17

process_2ac_aeqs, 17
process_2ac_ers, 18

quick_er_format, 18

range_splitter, 19
read_2a_sheets, 19
read_2aCmrkd, 20
read_2aCUandM, 20
read_2aCUandM_N, 21
read_2aCUnmrkd, 21
read_jdf, 22
read_jdf_unmarked, 22
read_key_tamm_sheets_chin, 23
read_key_tamm_sheets_coho, 23
read_limiting_stock, 24, *33*
read_management_chin, 24
read_overview, 25, *33*
read_overview_complete, 26
read_tnt_allocation_chin, 26
reformat_management, 27

safe_convert_numeric, 28

tamm_compare, 28
tamm_compare3, 29
tamm_diff, 30
tamm_format_input *(*tamm_format_sheets*)*, 31
tamm_format_limiting *(*tamm_format_sheets*)*, 31
tamm_format_overview *(*tamm_format_sheets*)*, 31
tamm_format_sheets, 31
tamm_format_wacoast *(*tamm_format_sheets*)*, 31
tamm_report, 32
trace_formula, 34

validate_data_frame, 35