Package: reportifyr (via r-universe)

October 25, 2024

Title Create reproducible reports with quarto and word

Version 0.2.0

Description A tool to extend quarto's ability to be a starting point for fully featured word reports.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 2.10)

Imports assertthat, cli, digest, dplyr, flextable, gert, ggplot2, here, jsonlite, log4r, officer, processx, purrr, rlang, rstudioapi, tictoc, yaml

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

LazyData true

VignetteBuilder knitr

URL https://a2-ai.github.io/reportifyr/

Repository https://a2-ai.r-universe.dev

RemoteUrl https://github.com/a2-ai/reportifyr

RemoteRef HEAD

RemoteSha f4474dfbcbaef4bc36ac3121c6fd66df6b6b60bd

Contents

add_footnotes *Inserts Footnotes in appropriate places in Word files*

Description

Reads in a .docx file and returns a new version with footnotes placed at appropriate places in the document.

Usage

```
add_footnotes(
  docx_in,
  docx_out,
  figures_path,
  tables_path,
  footnotes = NULL,
  debug = F
)
```

Arguments

docx_in	Path to the input .docx file
docx_out	Path to output .docx to save to
figures_path	Path to images and associated metadata directory
tables_path	Path to tables and associated metadata directory
footnotes	Path to standard_footnotes.yaml
debug	Debug

Examples

```
## Not run:

# ---------------------------------------------------------------------------
# Load all dependencies
# ---------------------------------------------------------------------------
docx_in <- file.path(here::here(), "report", "shell", "template.docx")
doc_dirs <- make_doc_dirs(docx_in = docx_in)
figures_path <- file.path(here::here(), "OUTPUTS", "figures")
tables_path <- file.path(here::here(), "OUTPUTS", "tables")
footnotes <- file.path(here::here(), "report", "standard_footnotes.yaml")

# ---------------------------------------------------------------------------
# Step 1.
# Table addition running add_tables will format and insert tables into the doc.
# ---------------------------------------------------------------------------
add_tables(
  docx_in = docx_in,
  docx_out = doc_dirs$doc_tables,
  tables_path = tables_path
)

# ---------------------------------------------------------------------------
# Step 2.
# Next we place in the plots using the add_plots function.
# ---------------------------------------------------------------------------
add_plots(
  docx_in = doc_dirs$doc_tables,
  docx_out = doc_dirs$doc_tabs_figs,
  figures_path = figures_path
)

# ---------------------------------------------------------------------------
# Step 3.
# Now we can add the footnotes to all the inserted figures and tables.
# ---------------------------------------------------------------------------
add_footnotes(
  docx_in = doc_dirs$doc_tabs_figs,
  docx_out = doc_dirs$doc_draft,
  figures_path = figures_path,
  tables_path = tables_path,
  footnotes = footnotes
)

## End(Not run)
```

add_plots *Inserts Figures in appropriate places in Word files*

Description

Reads in a .docx file and returns a new version with figures placed at appropriate places in the document.

Usage

```
add_plots(
  docx_in,
  docx_out,
  figures_path,
  fig_width = NULL,
  fig_height = NULL,
  debug = F
)
```

Arguments

`docx_in`	Path to the input .docx file
`docx_out`	Path to output .docx to save to
`figures_path`	Path to images file directory
`fig_width`	Figure width in inches. This is a global controller. Defaults to NULL. If NULL, the size is calculated calculated based on the pixels of the actual figure
`fig_height`	Figure height in inches. This is a global controller. Defaults to NULL. If NULL, the size is automatically calculated based on the pixels of the actual figure
`debug`	Debug

Examples

```
## Not run:

# ---------------------------------------------------------------------------
# Load all dependencies
# ---------------------------------------------------------------------------
docx_in <- file.path(here::here(), "report", "shell", "template.docx")
doc_dirs <- make_doc_dirs(docx_in = docx_in)
figures_path <- file.path(here::here(), "OUTPUTS", "figures")
tables_path <- file.path(here::here(), "OUTPUTS", "tables")
footnotes <- file.path(here::here(), "report", "standard_footnotes.yaml")

# ---------------------------------------------------------------------------
# Step 1.
# Table addition running add_tables will format and insert tables into the doc.
# ---------------------------------------------------------------------------
add_tables(
  docx_in = doc_dirs$doc_in,
  docx_out = doc_dirs$doc_tables,
  tables_path = tables_path
)
```

```
# ---------------------------------------------------------------------------
# Step 3.
# Next we place in the plots using the add_plots function.
# ---------------------------------------------------------------------------
add_plots(
  docx_in = doc_dirs$doc_tables,
  docx_out = doc_dirs$doc_tabs_figs,
  figures_path = figures_path
)

## End(Not run)
```

add_tables *Adds tables by looking for magic string*

Description

Adds tables by looking for magic string

Usage

```
add_tables(docx_in, docx_out, tables_path, debug = F)
```

Arguments

docx_in	input doc
docx_out	output doc
tables_path	path to tables file
debug	debug mode

Examples

```
## Not run:
# ---------------------------------------------------------------------------
# Load all dependencies
# ---------------------------------------------------------------------------
docx_in <- file.path(here::here(), "report", "shell", "template.docx")
doc_dirs <- make_doc_dirs(docx_in = docx_in)
figures_path <- file.path(here::here(), "OUTPUTS", "figures")
tables_path <- file.path(here::here(), "OUTPUTS", "tables")
footnotes <- file.path(here::here(), "report", "standard_footnotes.yaml")

# ---------------------------------------------------------------------------
# Step 1.
# Table addition running add_tables will format and insert tables into the doc.
# ---------------------------------------------------------------------------
add_tables(
  docx_in = doc_dirs$doc_clean,
  docx_out = doc_dirs$doc_tables,
  tables_path = tables_path
)

## End(Not run)
```

---

build_report *Updates a Word file to include formatted plots, tables, and footnotes*

Description

Reads in a .docx file and returns an updated version with plots, tables, and footnotes replaced.

Usage

```
build_report(
  docx_in,
  docx_out,
  figures_path,
  tables_path,
  standard_footnotes_yaml = NULL
)
```

Arguments

docx_in Path to input .docx to update

docx_out Path to output .docx to save to

figures_path Path to images file directory

tables_path Path to tables file directory

standard_footnotes_yaml
Path to standard_footnotes.yaml in report/

Examples

```
## Not run:

# ---------------------------------------------------------------------------
# Load all dependencies
# ---------------------------------------------------------------------------
docx_in <- file.path(here::here(), "report", "shell", "template.docx")
figures_path <- file.path(here::here(), "OUTPUTS", "figures")
tables_path <- file.path(here::here(), "OUTPUTS", "tables")
footnotes <- file.path(here::here(), "report", "standard_footnotes.yaml")

# ---------------------------------------------------------------------------
# Step 1.
# Run the wrapper function to replace figures, tables, and footnotes in a
# .docx file.
# ---------------------------------------------------------------------------
```

```
build_report(
  docx_in = docx_in,
  docx_out = doc_dirs$doc_draft,
  figures_path = figures_path,
  tables_path = tables_path,
  standard_footnotes_yaml = footnotes
)

## End(Not run)
```

`finalize_document` *Finalizes the document by removing magic strings and bookmarks*

Description

Reads in a .docx file and returns a finalized version with magic strings and bookmarks removed.

Usage

```
finalize_document(docx_in, docx_out = NULL)
```

Arguments

`docx_in`	Path to input .docx to finalize
`docx_out`	Path to output .docx to save to

Examples

```
## Not run:

# ---------------------------------------------------------------------------
# Load all dependencies
# ---------------------------------------------------------------------------
docx_in <- file.path(here::here(), "report", "shell", "template.docx")
figures_path <- file.path(here::here(), "OUTPUTS", "figures")
tables_path <- file.path(here::here(), "OUTPUTS", "tables")
footnotes <- file.path(here::here(), "report", "standard_footnotes.yaml")

# ---------------------------------------------------------------------------
# Step 1.
# Run the wrapper function build_report() to replace figures, tables, and
# footnotes in a .docx file.
# ---------------------------------------------------------------------------
build_report(
  docx_in = docx_in,
  docx_out = doc_dirs$doc_draft,
  figures_path = figures_path,
  tables_path = tables_path,
  standard_footnotes_yaml = footnotes
)
```

```
# ---------------------------------------------------------------------------
# Step 2.
# If you are happy with the report and are ready to finalize the document.
# ---------------------------------------------------------------------------
finalize_document(
  docx_in = doc_dirs$doc_draft,
  docx_out = doc_dirs$doc_final
)

## End(Not run)
```

format_flextable *Formats data frames to a flextable specification*

Description

Formats data frames to a flextable specification

Usage

```
format_flextable(data_in, table1_format = F)
```

Arguments

`data_in`	Data frame to be formatted
`table1_format`	Boolean for using table1 formatting

Value

A formatted flextable

Examples

```
## Not run:
dt <- head(iris, 10)
format_flextable(
  data_in = dt
)

## End(Not run)
```

get_meta_abbrevs *Get meta abbreviations from standard_footnotes.yaml*

Description

Get meta abbreviations from standard_footnotes.yaml

Usage

```
get_meta_abbrevs(path_to_footnotes_yaml)
```

Arguments

`path_to_footnotes_yaml`
The path to the standard_footnotes.yaml

Value

A list of meta_abbrevs to be called while performing an analysis

Examples

```
## Not run:
meta_abbrevs <- get_meta_abbrevs(
  path_to_footnotes_yaml = file.path(
    here::here(), "report", "standard_footnotes.yaml"
  )
)

## End(Not run)
```

get_meta_type *Get meta types from standard_footnotes.yaml*

Description

Get meta types from standard_footnotes.yaml

Usage

```
get_meta_type(path_to_footnotes_yaml)
```

Arguments

`path_to_footnotes_yaml`
The path to the standard_footnotes.yaml

Value

A list of meta_types to be called while performing an analysis

Examples

```
## Not run:
meta_type <- get_meta_type(
  path_to_footnotes_yaml = file.path(
    here::here(), "report", "standard_footnotes.yaml"
  )
)

## End(Not run)
```

ggsave_with_metadata *Wrapper around the ggplot2 ggsave function. Saves a ggplot (or other grid object) and captures analysis relevant metadata in a .json file*

Description

Extension to the ggsave function that allows capturing object metadata as a separate .json file.

Usage

```
ggsave_with_metadata(
  filename,
  plot = ggplot2::last_plot(),
  meta_type = "NA",
  meta_equations = NULL,
  meta_notes = NULL,
  meta_abbrevs = NULL,
  ...
)
```

Arguments

filename	File name to create on disk
plot	Plot to save, defaults to last plot displayed
meta_type	Parameter for specifying meta_type for write_object_metadata
meta_equations	Parameter for specifying additional equations
meta_notes	Parameter for specifying additional notes
meta_abbrevs	Parameter for specifying additional abbreviations
...	Additional args to be used in ggsave

Examples

```
## Not run:
# Path to the analysis figures (.png) and metadata (.json files)
figures_path <- file.path(tempdir(), "figures")

# ----------------------------------------------------------------------------
# Construct a simple ggplot
# ----------------------------------------------------------------------------
g <- ggplot2::ggplot(
  data = Theoph,
  ggplot2::aes(x = Time, y = conc, group = Subject)
) +
  ggplot2::geom_point() +
  ggplot2::geom_line() +
  ggplot2::theme_bw()

# Save a png using the helper function
out_name <- "01-12345-pk-timecourse1.png"
ggsave_with_metadata(
  filename = file.path(figures_path, out_name),
)

## End(Not run)
```

initialize_python *Initializes python virtual environment*

Description

Initializes python virtual environment

Usage

```
initialize_python()
```

Examples

```
## Not run:
initialize_python()

## End(Not run)
```

initialize_report_project *Create report directories within a project*

Description

Create report directories within a project

Usage

```
initialize_report_project(project_dir)
```

Arguments

project_dir	The path to the main directory folder

Examples

```
## Not run:
initialize_report_project(project_dir = tempdir())

## End(Not run)
```

make_doc_dirs *Helper function that defines document output paths*

Description

Helper function that defines document output paths

Usage

```
make_doc_dirs(docx_in)
```

Arguments

docx_in	Path to the input .docx file

Value

A list of document paths

Examples

```
## Not run:

# ---------------------------------------------------------------------------
# Load all dependencies
# ---------------------------------------------------------------------------
docx_in <- file.path(here::here(), "report", "shell", "template.docx")
doc_dirs  <- make_doc_dirs(docx_in = docx_in)

## End(Not run)
```

`preview_metadata` *Previews a single metadata file for an object*

Description

Previews a single metadata file for an object

Usage

```
preview_metadata(file_name)
```

Arguments

`file_name`	Path to an object to view its metadata

Value

A single row data frame consisting of metadata type and footnotes for the object supplied

Examples

```
## Not run:
preview_metadata("OUTPUTS/figures/myplot.png")

## End(Not run)
```

preview_metadata_files

Preview all metadata .json files in a directory

Description

Preview all metadata .json files in a directory

Usage

```
preview_metadata_files(file_dir)
```

Arguments

`file_dir`	Path to a directory containing metadata .json files

Value

A data frame of metadata footnotes and meta type

Examples

```
## Not run:
preview_metadata_file("OUTPUTS/figures/")

## End(Not run)
```

remove_tables_figures_footnotes

Removes Tables, Figures, and Footnotes from a Word file

Description

Reads in a .docx file and returns a new version with tables, figures, and footnotes removed from the document.

Usage

```
remove_tables_figures_footnotes(docx_in, docx_out)
```

Arguments

`docx_in`	Path to the input .docx file
`docx_out`	Path to output .docx to save to

Examples

```
## Not run:

# ----------------------------------------------------------------------------
# Load all dependencies
# ----------------------------------------------------------------------------
docx_in <- file.path(here::here(), "report", "shell", "template.docx")
doc_dirs <- make_doc_dirs(docx_in = docx_in)

# ----------------------------------------------------------------------------
# Removal to set-up docx_in
# ----------------------------------------------------------------------------
remove_tables_figures_footnotes(
  docx_in = docx_in,
  docx_out = doc_dirs$doc_clean
)

## End(Not run)
```

reportifyr *reportifyr: An R package to aid in the drafting of reports.*

Description

This pacakge aims to ease table, figure, and footnote insertion and formatting into reports.

reportifyr setup functions

- `initialize_report_project`: Creates `report` directory with shell, draft, scripts, final subdirectories and adds standard_footnotes.yaml to /report directory. Initializes python virtual environment through a subcall to initialize_python creates OUTPUTS/figures, OUTPUTS/tables, OUTPUTS/listings directories
- `initialize_python`: Creates virtual environment in options("venv_dir") if set or project root otherwise. Also installs python-docx and pyyaml packages.

Analysis output saving functions

- `ggsave_with_metadata`: Wrapper for saving ggplot that also creates metadata for plot object
- `save_rds_with_metadata`: Wrapper for saveRDS that also creates metadata for tabular object and saves as rtf
- `write_csv_with_metadata`: Wrapper for write.csv that also creates metadata for tabular object and saves as rtf
- `save_as_rtf`: Saves tabular object (.csv or flextable) as rtf. This is called within save_rds_with_metadata and write_csv_with_metadata
- `format_flextable`: Formats a tabular data object as a flextable with simple formatting.

metadata interaction functions

- write_object_metadata: Creates a metadata.json file for the input object path. Called within all analysis output saving functions.
- update_object_footnotes: Used to update footnotes fields within object metadata json files.
- preview_metadata_files: Generates a data frame of all object metadata within input directory and displays object, meta type, equations notes, abbreviations.
- preview_metadata: Generates the metadata data frame of the singular input file.
- get_meta_type: Generates meta_type object to allow user to see available meta_types in standard_footnotes.yaml within report directory
- get_meta_abbrevs: Generates meta_abbrev object to allow user to see available abbreviations in standard_footnotes.yaml within report directory.

Document interaction functions

- add_tables: Adds tables into the word document
- add_footnotes: Adds footnotes for figures and tables
- add_plots: Adds plots into the word document
- remove_tables_figures_footnotes: Removes all tables, figures, and footnotes associated with a magic string {rpfy}:object.ext
- remove_magic_strings: Removes all magic strings from a document cutting its tie to reportifyr but producing a final document.
- remove_bookmarks: Removes all bookmarks from document

Report building function

- build_report: Wrapper function to remove old tables, figures, and footnotes and adds new ones. Calls remove_tables_figures_footnotes, make_doc_dirs, add_plots, add_tables, add_footnotes
- finalize_document: Wrapper function to remove magic strings and bookmarks from document.

Utility functions

- make_doc_dirs: FILL IN
- validate_object: FILL IN

Author(s)

Maintainer: Jacob Dumbleton <jacob@a2-ai.com>

Authors:

- Matthew Smith <matthews@a2-ai.com>
- Theodoros Papathanasiou <theo.x.papathanasiou@gsk.com>
- Tarjinder Sahota <tarjinder.z.sahota@gsk.com>

- Devin Pastoor <devin@a2-ai.com>
- Agnes Kim <agnes@a2-ai.com>
- Anne Zheng <anne@a2-ai.com>
- Jenna Johnson <jenna@a2-ai.com>

Other contributors:

- Fenando Carreno <fernando.o.carreno@gsk.com> [contributor]
- Kashyap Patel <kashyap.x.patel@gsk.com> [contributor]
- Maxwell Chirehwa <maxwell.t.chirehwa@gsk.com> [contributor]
- Wesley Cummings <wes@a2-ai.com> [contributor]
- Elizabeth LeBeau <elizabeth@a2-ai.com> [contributor]

`save_rds_with_metadata` *Wrapper around the saveRDS function. Saves an object as .RDS and .RTF and captures analysis relevant metadata in a .json file*

Description

Extension to the saveRDS function that allows capturing object metadata as a separate .json file.

Usage

```
save_rds_with_metadata(
  object,
  file = "",
  meta_type = "NA",
  meta_equations = NULL,
  meta_notes = NULL,
  meta_abbrevs = NULL,
  table1_format = F,
  ...
)
```

Arguments

`object`	R object to serialize
`file`	A connection or the name of the file where the R object is saved to or read from
`meta_type`	The analysis meta type. Defaults to "NA"
`meta_equations`	Additional equations to add to metadata
`meta_notes`	Additional notes to add to metadata
`meta_abbrevs`	Additional abbrevs to add to metadata
`table1_format`	Boolean for declaring object is table1 format
`...`	Additional args to be used in saveRDS

Examples

```
## Not run:
# Path to the analysis tables (.RDS) and metadata (.json files)
tables.path <- file.path(tempdir(), "tables")

# ---------------------------------------------------------------------------
# Save a simple table
# ---------------------------------------------------------------------------

out_name <- "01-12345-pk-theoph.csv"
save_rds_with_metadata(
  object = Theoph,
  file = file.path(tables.path, out_name)
)

## End(Not run)
```

`toggle_logger` *Updates the logging level for functions. Default is set to WARN*

Description

Updates the logging level for functions. Default is set to WARN

Usage

```
toggle_logger()
```

Examples

```
## Not run:
Sys.setenv("RPFY_VERBOSE" = "DEBUG")
toggle_logger()

## End(Not run)
```

`update_object_footnotes` *Updates an object's footnote metadata - equations, notes, or abbreviations*

Description

Updates an object's footnote metadata - equations, notes, or abbreviations

Usage

```
update_object_footnotes(
  file_path,
  overwrite = FALSE,
  equations = NULL,
  notes = NULL,
  abbrevs = NULL
)
```

Arguments

`file_path`	Path to object or object's metadata file
`overwrite`	Boolean to overwrite existing entries or append. Default is to append
`equations`	String or vector of strings of equations to add
`notes`	String or vector of strings of notes to add
`abbrevs`	String or vector of strings of abbreviations to add

Examples

```
## Not run:
update_object_footnotes("example_metadata.json", equations = c("K10 = CL/VC", "K12 = Q/VC"))

## End(Not run)
```

`validate_object` *Validates a file's hash against a stored hash in the associated _metadata.json file*

Description

Validates a file's hash against a stored hash in the associated _metadata.json file

Usage

```
validate_object(file)
```

Arguments

`file`	A connection or the name of the file where the R object is saved to or read from

Value

A boolean declaring if the hashes are equal or not

Examples

```
## Not run:
tables.path  <- "OUTPUTS/tables"
out_name <- "01-12345-pk-theoph.csv"

validate_object(file = file.path(tables.path, out_name))

## End(Not run)
```

write_csv_with_metadata

Wrapper around the write.csv function. Saves data as .RDS and .RTF and captures analysis relevant metadata in a .json file

Description

Extension to the write.csv function that allows capturing object metadata as a separate .json file.

Usage

```
write_csv_with_metadata(
  object,
  file,
  meta_type = "NA",
  meta_equations = NULL,
  meta_notes = NULL,
  meta_abbrevs = NULL,
  table1_format = F,
  ...
)
```

Arguments

`object` The object to be written, preferably a matrix or data frame. If not, it is attempted to coerce object to a data frame

`file` Either a character string naming a file or a connection open for writing. "" indicates output to the console.

`meta_type` The analysis meta type. Defaults to "NA"

`meta_equations` Additional equations for metadata

`meta_notes` Additional notes for metadata

`meta_abbrevs` Additional abbrevs for metadata

`table1_format` Boolean for declaring table is table1 format

`...` Additional arguments that can be passed to write.csv

Examples

```
## Not run:

# Path to the analysis tables (.csv) and metadata (.json files)
tables.path <- "OUTPUTS/tables"

# ------------------------------------------------------------------------------
# Save a simple table
# ------------------------------------------------------------------------------

out_name <- "01-12345-pk-theoph.csv"
write_csv_with_metadata(
  object = Theoph,
  file = file.path(tables.path, out_name), row_names = F
)

## End(Not run)
```

write_object_metadata *Writes an object's metadata .json file*

Description

Writes an object's metadata .json file

Usage

```
write_object_metadata(
  object_file,
  meta_type = NULL,
  equations = NULL,
  notes = NULL,
  abbrevs = NULL,
  table1_format = F
)
```

Arguments

object_file	Path to the file of the object to write metadata for
meta_type	A string denoting what standard notes to use
equations	Additional equations to include in metadata, either string of single equation or vector of multiple
notes	Additional notes to include in metadata, either string of single note or vector of multiple
abbrevs	Additional abbreviations to include in metadata, either string of single abbrev or vector of multiple
table1_format	Boolean for using table1 formatting for flextables

Examples

```
## Not run:
ft <- flextable(iris)

write_object_metadata(ft, "table", file_path)

## End(Not run)
```

Index