Package 'saros.base'

May 9, 2026

Type Package

Title Base Tools for Semi-Automatic Reporting of Ordinary Surveys

Version 1.2.0

Maintainer Stephan Daus <stephus.daus@gmail.com>

Description Scaffold an entire web-based report using template chunks, based on a small chapter overview and a dataset. Highly adaptable with prefixes, suffixes, translations, etc. Also contains tools for password-protecting, e.g. for each organization's report on a website. Developed for the common case of a survey across multiple organizations/sites where each organization wants to obtain results for their organization compared with everyone else. See 'saros' (<https://CRAN.R-project.org/package=saros>) for tools used for authors in the drafted reports.

Note Free to use for non-Norwegian institutions, otherwise see LICENSE.

License MIT + file LICENSE

URL https://nifu-no.github.io/saros.base/, https://github.com/NIFU-NO/saros.base

BugReports https://github.com/NIFU-NO/saros.base/issues

Depends R (>= 4.2.0)

Imports stats, utils, vctrs, rlang, cli, tidyselect, dplyr, tidyr, glue, stringi, forcats, fs, yaml, zip, rstudioapi, bcrypt

Suggests covr, haven, srvyr, readr, qs, purrr, writexl, webshot, usethis, quarto, labelled, testthat (>= 3.0.0), tibble, withr, spelling

SystemRequirements None.

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Language en-US

VignetteBuilder quarto

Config/Needs/website rmarkdown

Config/testthat/parallel true

NeedsCompilation no

Author Stephan Daus [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0003-0230-6997>),
Nordic Institute for The Studies of Innovation, Research and Education
(NIFU) [fnd],
Kristiania University College [fnd]

Repository CRAN

Date/Publication 2025-11-12 18:50:02 UTC

Contents

`check_variable_labels` *Check Variable Labels for Saros Use*

Description

Checks various label quality issues and returns a data frame with indications of which label has the issue and what issue.

Usage

```
check_variable_labels(
  data,
  separator = " - ",
  special_chars = "[^\\p{L}\\p{N}\\s!?'#%&/()\\[\\]{}=+\\-*.,:;]"
)
```

Arguments

`data`	Data frame or tibble.
`separator`	String, indicating what to check that there is maximum of 1 of per label.
`special_chars`	String of regular expression.

Value

Data frame

Examples

```
df <- data.frame(
  a = 1:3,
  b = 4:6,
  c = 7:9,
  d = 11:13,
  e = 14:16,
  f = 17:19
)
attr(df$a, "label") <- "Age"
attr(df$b, "label") <- "Age"
attr(df$c, "label") <- "Gender - Male"
attr(df$e, "label") <- "Gender - Male -  2"
attr(df$f, "label") <- "Gender - Female..."
check_variable_labels(df)
```

copy_folder_contents_to_dir

Convenience Function to Copy Only the Contents of A Folder to Another Folder

Description

Convenience Function to Copy Only the Contents of A Folder to Another Folder

Usage

```
copy_folder_contents_to_dir(
  from,
  to,
  overwrite = FALSE,
  only_copy_folders = FALSE
)
```

Arguments

to, from	String, path from where to copy the contents, and where to copy them to.
overwrite	Flag. Defaults to FALSE.
only_copy_folders	Flag. Defaults to FALSE. If TRUE, only copies folders.

Value

No return value, called for side effects

Examples

```
copy_folder_contents_to_dir(
  from = system.file("help", "figures", package = "dplyr"),
  to = tempdir()
)
```

create_directory_structure

Create a Pre-defined Directory Hierarchy on Disk

Description

Create a Pre-defined Directory Hierarchy on Disk

Usage

```
create_directory_structure(
  path,
 structure_path = system.file("templates", "_project_structure_en.yaml", package =
    "saros.base"),
  numbering_prefix = c("none", "max_local", "max_global"),
  numbering_inheritance = TRUE,
  word_separator = NULL,
  numbering_parent_child_separator = word_separator,
  numbering_name_separator = " ",
  case = c("asis", "sentence", "title", "lower", "upper", "snake"),
  replacement_list = c(project_initials = "SSN"),
  create = FALSE,
  count_existing_folders = FALSE
)
```

Arguments

path	String, path to where to create the project files
structure_path	String. Path to the YAML file that defines the folder structure. Defaults to system.file("templates", "_project_structure_en.yaml").
numbering_prefix	String. One of c("none", "max_local", "max_global").
numbering_inheritance	Flag. Whether to inherit numbering from parent folder.
word_separator	String. Replace separators between words in folder names. Defaults to NULL.
numbering_parent_child_separator	String. Defaults to word_separator.
numbering_name_separator	String. Separator between numbering part and name.
case	String. One of c("asis", "sentence", "lower", "upper", "title", "snake").
replacement_list	named character vector. Each name in this vector will be replaced with its "{{value}}" in the structure_path file
create	Boolean. Defaults to TRUE in initialize_saros_project(), FALSE in create_directory_structure().
count_existing_folders	Boolean. Defaults to FALSE.

Value

No return value, called for side effects

Examples

```
struct <- create_directory_structure(path = tempdir(), create = FALSE)
```

`create_email_credentials` *Create Data Frame Containing Email Drafts with User Credentials*

Description

Create Data Frame Containing Email Drafts with User Credentials

Usage

```
create_email_credentials(
  email_data_frame,
  email_col = "email",
  username_col = "username",
  local_main_password_path = ".htpasswd_private",
  ignore_missing_emails = FALSE,
 email_body = "Login credentials are \nUsername: {username},\nPassword: {password}",
  email_subject = "User credentials for website example.net.",
  ...
)
```

Arguments

`email_data_frame`	Data.frame/tibble with (at least) emails and usernames
`email_col`	String, name of email column
`username_col`	String, name of username column in email_data_frame
`local_main_password_path`	Path to a local .htpasswd file containing username:password header and : as separator.
`ignore_missing_emails`	Flag, defaults to FALSE. Whether usernames existing in password file but not email file will result in warnings.
`email_body, email_subject`	String, subject line and email body respectively. Supports glue syntax referring to columns found in the email data frame or password file.
`...`	Dynamic dots forwarded to quarto::quarto_render

Value

Data.frame

`create_r_files` *Create Folder with Placeholder R-files Based on Structure in CSV-file*

Description

Create Folder with Placeholder R-files Based on Structure in CSV-file

Usage

```
create_r_files(
  r_files_out_path,
 r_files_source_path = system.file("templates", "r_files.csv", package = "saros.base"),
  r_optionals = TRUE,
  r_add_file_scope = TRUE,
  r_prefix_file_scope = "### ",
  r_add_folder_scope_as_README = FALSE,
  word_separator = NULL,
  case = c("asis", "sentence", "title", "lower", "upper", "snake"),
  numbering_prefix = c("none", "max_local", "max_global"),
  numbering_inheritance = TRUE,
  numbering_parent_child_separator = word_separator,
  numbering_name_separator = " "
)
```

Arguments

`r_files_out_path`
: String, path to where to place R placeholder files. If NULL, will not create any.

`r_files_source_path`
: String, path to where to find CSV-field containing the columns folder_name, folder_scope, file_name, file_scope. If NULL, defaults to system.file("templates", "r_files.csv")).

`r_optionals`
: Flag. Whether to add files marked as 1 (or TRUE) in the optional column. Defaults to TRUE.

`r_add_file_scope`
: Flag. Whether to add value from column 'file_scope' to beginning of each file. Default to TRUE.

`r_prefix_file_scope`
: String to add before file_scope. Defaults to "### "

`r_add_folder_scope_as_README`
: Flag. Whether to create README file in each folder with the folder_scope column cell in r_files_source_path. Defaults to FALSE.

`word_separator`
: String. Replace separators between words in folder names. Defaults to NULL.

`case`
: String. One of c("asis", "sentence", "lower", "upper", "title", "snake").

`numbering_prefix`
: String. One of c("none", "max_local", "max_global").

`numbering_inheritance`
: Flag. Whether to inherit numbering from parent folder.

`numbering_parent_child_separator`
: String. Defaults to word_separator.

`numbering_name_separator`
: String. Separator between numbering part and name.

Value

No return value, called for side effects

Examples

```
create_r_files(r_files_out_path = tempdir())
```

---

detect_malformed_quarto_project

Detect malformed Quarto website projects generated by saros.base

---

Description

This function checks for common issues in a Quarto website project directory. Currently, it finds:

- Subfolders lacking index.qmd
- .qmd files without a title in their YAML front matter Future versions will add more checks.

Usage

```
detect_malformed_quarto_project(root_dir)
```

Arguments

`root_dir`	Path to the root directory of the Quarto website project.

Value

A data.frame with columns: type, path, details

Examples

```
detect_malformed_quarto_project(tempdir())
```

download_zip_to_folder *Wrapper to Download and Unzip a Github Repository to A Folder*

Description

Wrapper to Download and Unzip a Github Repository to A Folder

Usage

```
download_zip_to_folder(
 github_zip_url = "https://github.com/NIFU-NO/nifutemplates/archive/refs/heads/main.zip",
  zip_path = tempfile(fileext = ".zip"),
  files = NULL,
  out_path,
  prompt = TRUE,
  overwrite = FALSE,
  open_project = FALSE,
  newSession = TRUE
)
```

Arguments

github_zip_url	URL to zip file, as string.
zip_path	String, where to store zip-file. Defaults to a temporary location.
files	Character vector of files in zip-file to include. See `zip::unzip()`.
out_path	String, directory to where to store the unzipped files.
prompt	Flag, whether to ask user if conflicting files should be overwritten, if any. Defaults to TRUE.
overwrite	Flag, whether to overwrite files in out_path. Defaults to FALSE.
open_project	Flag or string. If FALSE (default), does nothing. If TRUE (requires `rstudioapi`-pkg), opens an assumed .Rproj-file in out_path after copying, or gives warning if not found. Alternatively, a string (path) can be provided. Defaults to file.path(out_path, ".Rproj") if such exists. Set to NULL or FALSE to ignore.
newSession	Flag. Whether to open new project in a new RStudio session. Defaults to TRUE.

Value

Character vector of unzipped files.

Examples

```
download_zip_to_folder(
 github_zip_url = "https://github.com/NIFU-NO/nifutemplates/archive/refs/heads/main.zip",
  out_path = tempdir(), overwrite = TRUE
)
```

draft_report *Automatically Draft a Quarto Report*

Description

The `draft_report()` function takes a raw dataset (`data`-argument) and the output from the `refine_chapter_overview()`-function as the `chapter_structure`-argument and outputs a set of pre-populated qmd-files in the specified `path`-folder. You can edit, render, and ultimately publish these as usual with Quarto features in RStudio. See also `{saros.post}`-package for post-processing tools.

Usage

```
draft_report(
  data,
  chapter_structure,
  ...,
  path = tempdir(),
  title = NULL,
  authors = NULL,
  authors_col = "author",
  chapter_yaml_file = NULL,
  chapter_qmd_start_section_filepath = NULL,
  chapter_qmd_end_section_filepath = NULL,
  index_filename = "index",
  index_yaml_file = NULL,
  index_qmd_start_section_filepath = NULL,
  index_qmd_end_section_filepath = NULL,
  report_filename = "report",
  report_yaml_file = NULL,
  report_qmd_start_section_filepath = NULL,
  report_qmd_end_section_filepath = NULL,
  report_includes_files = FALSE,
  ignore_heading_for_group = c(".template_name", ".variable_type_dep",
    ".variable_type_indep", ".variable_group_dep", "chapter"),
 replace_heading_for_group = c(chapter = ".chapter_number", .variable_label_suffix_dep =
  ".variable_name_dep", .variable_label_suffix_indep = ".variable_name_indep"),
  prefix_heading_for_group = NULL,
  suffix_heading_for_group = NULL,
  require_common_categories = TRUE,
  combined_report = TRUE,
  write_qmd = TRUE,
  attach_chapter_dataset = TRUE,
  auxiliary_variables = NULL,
  serialized_format = c("rds", "qs"),
  max_path_warning_threshold = 260,
  filename_prefix = "",
  data_filename_prefix = "data_",
```

```
  report_includes_prefix = "{{< include \"",
  report_includes_suffix = "\" >}}",
  log_file = NULL
)
```

Arguments

data	*Survey data* obj:<data.frame>|obj:<tbl_df>|obj:<srvyr> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_structure 'dep', etc columns.
chapter_structure	*What goes into each chapter and sub-chapter* obj:<data.frame>|obj:<tbl_df> // Required Data frame (or tibble, possibly grouped). One row per chapter. Should contain the columns 'chapter' and 'dep', Optionally 'indep' (independent variables) and other informative columns as needed.
...	*Dynamic dots* <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.
path	*Output path* scalar<character> // *default:* tempdir() (optional) Path to save all output. Defaults to a temporary directory.
title	*Title of report* scalar<character> // *default:* NULL (optional) Added automatically to YAML-header of index.qmd and report.qmd-files.
authors	*Authors of entire report* vector<character> // *default:* NULL (optional) If NULL, infers from chapter_structure[[authors_col]], and collates for entire report. If multiple authors per chapter, separate with semicolon. Ensure consistency.
authors_col	*Column name for author* scalar<character> // *default:* "author" (optional) Only used if it exists. Multiple authors are separated by semicolon (and optionally with a subsequent space).
chapter_yaml_file	*Path to YAML-file to insert into each chapter qmd-file* scalar<character> // *default:* NULL (optional) Path to file used to insert header YAML, in each chapter.
chapter_qmd_start_section_filepath, chapter_qmd_end_section_filepath, index_qmd_start_section_filepath, index_qmd_end_section_filepath, report_qmd_start_section_filepath, report_qmd_end_section_filepath	*Path to qmd-bit for start/end of each qmd* scalar<character> // *default:* NULL (optional) Path to qmd-snippet placed before/after body of all chapter/index/report qmd-files.

index_filename *Index filename*

scalar<character> // *default:* "index" (optional)

The name of the main index Quarto file used as landing page for each report. Will link to a PDF (report.qmd) which collects all chapters.

index_yaml_file, report_yaml_file

Path to YAML-file to insert into index.qmd and report.qmd respectively

scalar<character> // *default:* NULL (optional)

Path to file used to insert header YAML, in index and report files.

report_filename

Report filename

scalar<character> // *default:* "report" (optional)

The name of the main report QMD-file used when compiling a complete report collecting all chapters in its folder (except itself). If provided, will be linked to in the index. If NULL, will generate a filename based on the report title, prefixed with "0_". To turn off, set pdf=FALSE.

report_includes_files

Whether report.qmd includes {{< include 'chapter.qmd' >}}

scalar<logical> // *default:* FALSE

Useful to have in mesos reports. However, bear in mind that including other qmd files with conflicting YAML-headers might be risky.

ignore_heading_for_group

Ignore heading for group

vector<character> // *default:* NULL (optional)

Type of refined chapter_structure data for which to suppress the heading in the report output. Typically variable_name_dep, variable_name_indep, etc.

replace_heading_for_group

Replacing heading for group

named vector<character> // *default:* c(".variable_label_suffix_dep" = ".variable_name_dep")

Occasionally, one needs to replace the heading with another piece of information in the refined chapter_structure. For instance, one may want to organize output by variable_name_indep, but to display the variable_label_indep instead. Use the name for the replacement and the value for the original.

prefix_heading_for_group, suffix_heading_for_group

Prefix and suffix headings

vector<named character> // *default:* NULL (optional)

Names are heading_groups, values are the prefixes and suffixes. Note that prefixes should end with a \n as headings must begin on a new line.

require_common_categories

Check common categories

scalar<logical> // *default:* NULL (optional)

Whether to check if all items share common categories.

combined_report

Create a combined report?

scalar<logical> // *default:* FALSE (optional)

Whether to create a qmd file that merges all chapters into a combined report.

`write_qmd` *Toggle whether to make qmd-files*
`scalar<logical>` // *default:* `TRUE`

Sometimes it is useful to only create chapter_dataset files if these have been updated, without having to overwrite the qmd files.

`attach_chapter_dataset`
Toggle inclusion of chapter-specific datasets in qmd-files
`scalar<logical>` // *default:* `FALSE`

Whether to save in each chapter folder an 'Rds'-file with the chapter-specific dataset, and load it at the top of each QMD-file.

`auxiliary_variables`
Auxiliary variables to be included in datasets
`vector<character>` // *default:* `NULL` (`optional`)

Column names in `data` that should always be included in datasets for chapter qmd-files, if `attach_chapter_dataset=TRUE`. Not publicly available.

`serialized_format`
Serialized format
`scalar<string>` // *default:* `"rds"`

Format for serialized data when storing chapter dataset. One of `"rds"` (default), `"qs"` or `"fst"`. The latter two requires the respective packages to be installed. `"qs"` is usually the fastest and most space efficient, but sets package dependencies on the report project.

`max_path_warning_threshold`
Maximum number of characters in paths warning
`scalar<integer>` // *default:* `260` (`optional`)

Microsoft has set an absolute limit of 260 characters for its Sharepoint/OneDrive file paths. This will mean that files with cache (hash suffixes are added) will quickly breach this limit. When set, a warning will be returned if files are found to be longer than this threshold. Also note that spaces count as three characters due to its URL-conversion: %20. To avoid test, set to Inf

`filename_prefix`
Prefix string for all qmd filenames
`scalar<character>` // *default:* `""` (`optional`)

For mesos setup it might be useful to set these files (and related sub-folders) with an underscore (`filename_prefix = "_"`) in front as other stub files will include these main qmd files.

`data_filename_prefix`
String attached to beginning of data-file and data-object
`scalar<string>` // *default:* `"data_"`

`report_includes_prefix`, `report_includes_suffix`
Strings around files in report.qmd
`scalar<string>` // *default:* `"\{\{< include " and " >\}\}"`

The prefix and suffix for each of the chapters being included in the report.qmd file if `report_includes_files = TRUE`.

`log_file` *Path to log file*
`scalar<string>` // *default:* `"_log.txt"` (`optional`)

Path to log file. Set to NULL to disable logging.

Details

Note that saros treats data as they are stored: numeric, integer, factor, ordinal, character, and date-time. Currently, only factor/ordinal and character are implemented.

Value

The path-argument.

Examples

```
ex_survey_ch_structure <-
  refine_chapter_overview(
    chapter_overview = ex_survey_ch_overview,
    data = ex_survey
  )
index_filepath <-
  draft_report(
    chapter_structure = ex_survey_ch_structure,
    data = ex_survey,
    path = tempdir()
  )
```

---

ex_survey *ex_survey: Mockup dataset of a survey.*

Description

A dataset containing fake respondents' answers to survey questions. The first two, x_sex and x_human, are intended to be independent variables, whereas the remaining are dependent. The underscore _ in variable names separates item groups (prefix) from items (suffix) (i.e. a_1-a_9 => a + 1-9), whereas ' - ' separates the same for labels. The latter corresponds with the default in SurveyXact.

Usage

```
ex_survey
```

Format

A data frame with 100 rows and 29 variables:

x1_sex Gender

x2_human Is respondent human?

x3_nationality Where is the respondent born?

a_1 Do you consent to the following? - Agreement #1

a_2 Do you consent to the following? - Agreement #2

a_3 Do you consent to the following? - Agreement #3

a_4 Do you consent to the following? - Agreement #4

a_5 Do you consent to the following? - Agreement #5

a_6 Do you consent to the following? - Agreement #6

a_7 Do you consent to the following? - Agreement #7

a_8 Do you consent to the following? - Agreement #8

a_9 Do you consent to the following? - Agreement #9

b_1 How much do you like living in - Beijing

b_2 How much do you like living in - Brussels

b_3 How much do you like living in - Budapest

c_1 How many years of experience do you have in - Company A

c_2 How many years of experience do you have in - Company B

d_1 Rate your degree of confidence doing the following - Driving

d_2 Rate your degree of confidence doing the following - Drinking

d_3 Rate your degree of confidence doing the following - Driving

d_4 Rate your degree of confidence doing the following - Dancing

e_1 How often do you do the following? - Eat

e_2 How often do you do the following? - Eavesdrop

e_3 How often do you do the following? - Exercise

e_4 How often do you do the following? - Encourage someone whom you have only recently met and who struggles with simple tasks that they cannot achieve by themselves

p_1 To what extent do you agree or disagree to the following policies - Red Party

p_2 To what extent do you agree or disagree to the following policies - Green Party

p_3 To what extent do you agree or disagree to the following policies - Yellow Party

p_4 To what extent do you agree or disagree to the following policies - Blue Party

f_uni Which of the following universities would you prefer to study at?

open_comments Do you have any comments to the survey?

resp_status Response status

`ex_survey_ch_overview` *ex_survey_ch_overview: Mock overview of chapter structure*

Description

Note that only chapter and dep are compulsory.

Usage

```
ex_survey_ch_overview
```

Format

A data frame with 5 rows (chapters) and 5 variables:

chapter Manual entry chapter title

author Single, or multiple authors separated by semicolon

dep Columns in ex_survey having the role of dependent variable

indep Columns in ex_survey having the role of independent variable

irrelevant_col Just a column about something else to verify that the system works also with superfluous information.

`filename_sanitizer` *File/folder name sanitizer replacing space and punctuation with underscore*

Description

File/folder name sanitizer replacing space and punctuation with underscore

Usage

```
filename_sanitizer(
  x,
  max_chars = NA_integer_,
  accept_hyphen = FALSE,
  sep = "_",
  valid_obj = FALSE,
  to_lower = FALSE,
  make_unique = TRUE
)
```

Arguments

`x`	Character vector of file/folder names
`max_chars`	Maximum character length
`accept_hyphen`	Flag, whether a hyphen - is acceptable.
`sep`	String, replacement for illegal characters and spaces.
`valid_obj`	Flag, whether output should be valid as R object name.
`to_lower`	Flag, whether to force all characters to lower.
`make_unique`	Flag, whether all should be unique.

Value

Character vector of same length as x

Examples

```
filename_sanitizer(c("Too long a name", "with invalid *^/&#"))
```

generate_yaml_from_directory

Generate YAML File from Directory Structure

Description

Generate YAML File from Directory Structure

Usage

```
generate_yaml_from_directory(
  input_path = tempdir(),
  output_yaml_path = "_project_structure_en.yaml",
  remove_prefix_numbers = FALSE
)
```

Arguments

input_path	String. The path to the directory whose structure needs to be captured.
output_yaml_path	String. The path where the YAML file will be saved.
remove_prefix_numbers	Boolean. Whether to remove numeric prefixes and any resulting leading non-alphanumeric characters from folder names. Defaults to FALSE.

Value

No return value, called for side effects

Examples

```
generate_yaml_from_directory(
  output_yaml_path =
    tempfile("_project_structure_en", fileext = ".yaml")
)
```

get_chunk_template_defaults

Get Global Options for Chunk Templates

Description

Get Global Options for Chunk Templates

Usage

```
get_chunk_template_defaults(variant = 1)
```

Arguments

variant	Positive integer.

Value

List with options in R

Examples

```
get_chunk_template_defaults()
```

---

get_organize_by_opts *Get Core Chapter Structure Column Names*

Description

Returns the vector of core column names available as organize_by options.

Usage

```
get_organize_by_opts()
```

Value

A character vector.

Examples

```
get_organize_by_opts()
```

get_raw_labels *Helper function to extract raw variable labels from the data*

Description

Helper function to extract raw variable labels from the data

Usage

```
get_raw_labels(data, col_pos = NULL, return_as_list = FALSE)
```

Arguments

data	Dataset
col_pos	Optional, character vector of column names or integer vector of positions
return_as_list	Flag, whether to return as list or character vector

Value

List or character vector

initialize_saros_project *Initialize Folder Structure*

Description

Can be used programatically from the console, or simply use the New Project Wizard.

Usage

```
initialize_saros_project(
  path,
  structure_path = NULL,
  numbering_prefix = c("none", "max_local", "max_global"),
  numbering_inheritance = TRUE,
  word_separator = NULL,
  numbering_name_separator = " ",
  replacement_list = NULL,
  numbering_parent_child_separator = word_separator,
  case = c("asis", "sentence", "title", "lower", "upper", "snake"),
  count_existing_folders = FALSE,
  r_files_out_path = NULL,
  r_files_source_path = system.file("templates", "r_files.csv", package = "saros.base"),
  r_optionals = TRUE,
```

```
  r_add_file_scope = TRUE,
  r_prefix_file_scope = "### ",
  r_add_folder_scope_as_README = FALSE,
  create = TRUE
)
```

Arguments

path	String, path to where to create the project files
structure_path	String. Path to the YAML file that defines the folder structure. Defaults to system.file("templates", "_project_structure_en.yaml").
numbering_prefix	String. One of c("none", "max_local", "max_global").
numbering_inheritance	Flag. Whether to inherit numbering from parent folder.
word_separator	String. Replace separators between words in folder names. Defaults to NULL.
numbering_name_separator	String. Separator between numbering part and name.
replacement_list	named character vector. Each name in this vector will be replaced with its "{{value}}" in the structure_path file
numbering_parent_child_separator	String. Defaults to word_separator.
case	String. One of c("asis", "sentence", "lower", "upper", "title", "snake").
count_existing_folders	Boolean. Defaults to FALSE.
r_files_out_path	String, path to where to place R placeholder files. If NULL, will not create any.
r_files_source_path	String, path to where to find CSV-field containing the columns folder_name, folder_scope, file_name, file_scope. If NULL, defaults to system.file("templates", "r_files.csv")).
r_optionals	Flag. Whether to add files marked as 1 (or TRUE) in the optional column. Defaults to TRUE.
r_add_file_scope	Flag. Whether to add value from column 'file_scope' to beginning of each file. Default to TRUE.
r_prefix_file_scope	String to add before file_scope. Defaults to "### "
r_add_folder_scope_as_README	Flag. Whether to create README file in each folder with the folder_scope column cell in r_files_source_path. Defaults to FALSE.
create	Boolean. Defaults to TRUE in initialize_saros_project(), FALSE in create_directory_structure().

Value

Returns invisibly path

Examples

```
initialize_saros_project(path = tempdir())
```

`is_string` *Is x A String?*

Description

Returns TRUE if object is a character of length 1.

Usage

```
is_string(x)
```

Arguments

x	Object

Value

Bool

`read_default_draft_report_args` *Read Default Arguments for* `draft_report()` *from YAML-file*

Description

Read Default Arguments for `draft_report()` from YAML-file

Usage

```
read_default_draft_report_args(path)
```

Arguments

path	scalar<character> // Required. *default:* `settings.yaml`

Value

The defaults as a `yaml`-object.

Examples

```
tmpfile <- tempfile(fileext = ".yaml")
write_default_draft_report_args(path = tmpfile)
read_default_draft_report_args(path = tmpfile)
```

refine_chapter_overview *Processes A 'chapter_overview' Data Frame*

Description

Processes A 'chapter_overview' Data Frame

Usage

```
refine_chapter_overview(
  chapter_overview = NULL,
  data = NULL,
  chunk_templates = NULL,
  label_separator = " - ",
  name_separator = NULL,
  single_y_bivariates_if_indep_cats_above = 3,
  single_y_bivariates_if_deps_above = 20,
  always_show_bi_for_indep = NULL,
  hide_bi_entry_if_sig_above = 1,
  hide_chunk_if_n_below = 10,
  hide_variable_if_all_na = TRUE,
  keep_dep_indep_if_no_overlap = FALSE,
  organize_by = c(".chapter_number", ".variable_label_prefix_dep",
    ".variable_name_indep", ".template_name"),
 arrange_section_by = c(.chapter_number = FALSE, chapter = FALSE, .variable_position_dep
    = FALSE, .variable_position_indep = FALSE, .template_name = FALSE),
  na_first_in_section = TRUE,
  max_width_obj = 128,
  max_width_chunk = 128,
  max_width_file = 64,
  max_width_folder_name = 12,
  sep_obj = "_",
  sep_chunk = "-",
  sep_file = "-",
  filename_prefix = "",
  ...,
  progress = TRUE,
  variable_group_dep = ".variable_group_dep",
  variable_group_prefix = NULL,
  n_range_glue_template_1 = "{n}",
  n_range_glue_template_2 = "[{n[1]}-{n[2]}]",
  log_file = NULL
)
```

Arguments

chapter_overview

What goes into each chapter and sub-chapter
obj:<data.frame>|obj:<tbl_df> // Required

Data frame (or tibble, possibly grouped). One row per chapter. Should contain the columns 'chapter' and 'dep', Optionally 'indep' (independent variables) and other informative columns as needed.

data

Survey data
obj:<data.frame>|obj:<tbl_df>|obj:<srvyr> // Required

A data frame (or a srvyr-object) with the columns specified in the chapter_structure 'dep', etc columns.

chunk_templates

Chunk templates
obj:<data.frame>|obj:<tbl_df>|NULL // *default:* NULL (optional)

Must contain columns name (user-specified unique name for the template), template (the chunk template as {glue}-specification, variable_type_dep and optionally variable_type_indep. The latter two are list-columns of prototype vectors specifying which data the template will be applied to. Can optionally contain columns whose names match the default options for the function. These will then override the default function-wide options for the specific template.

label_separator

Variable label separator
scalar<character> // *default:* NULL (optional)

String to split labels on main question and sub-items.

name_separator *Variable name separator*
scalar<character> // *default:* NULL (optional)

String to split column names in data between main question and sub-items

single_y_bivariates_if_indep_cats_above

Single y bivariates if indep-cats above ...
scalar<integer> // *default:* 3 (optional)

Figures and tables for bivariates can become very long if the independent variable has many categories. This argument specifies the number of indep categories above which only single y bivariates should be shown.

single_y_bivariates_if_deps_above

Single y bivariates if dep-vars above ...
scalar<integer> // *default:* 20 (optional)

Figures and tables for bivariates can become very long if there are many dependent variables in a battery/question matrix. This argument specifies the number of dep variables above which only single y bivariates should be shown. Set to 0 to always show single y bivariates.

always_show_bi_for_indep

Always show bivariate for indep-variable
vector<character> // *default:* NULL (optional)

Specific combinations with a by-variable where bivariates should always be shown.

hide_bi_entry_if_sig_above

p-value threshold for hiding bivariate entry

scalar<double> // *default:* 1 (optional)

Whether to hide bivariate entry if significance is above this value. Defaults to showing all.

hide_chunk_if_n_below

Hide result if N below

scalar<integer> // *default:* 10 (optional)

Whether to hide result if N for a given dataset is below this value. NOTE: Exceptions will be made to chr_table and chr_plot as these are typically exempted in the first place. This might change in the future with a separate argument.

hide_variable_if_all_na

Hide variable from outputs if containing all NA

scalar<boolean> // *default:* TRUE (optional)

Whether to remove variables if all values are NA.

keep_dep_indep_if_no_overlap

Keep dep-indep if no overlap

scalar<boolean> // *default:* FALSE (optional)

Whether to keep dep-indep rows if there is no overlap.

organize_by

Grouping columns

vector<character> // *default:* NULL (optional)

Column names used for identifying chapters and sections.

arrange_section_by

Sorting columns

vector<character> or named vector<logical> // *default:* NULL (optional)

Column names used for sorting sections within each organize_by group. Can include any column present in the output dataframe (both original and generated columns). If character vector, will assume all are to be arranged in ascending order. If a named logical vector, FALSE will indicate ascending, TRUE descending. An error will be thrown if any specified column does not exist in the output. Defaults to sorting in ascending order (alphabetical) for commonly needed variable name/label info, and in descending order for chunk_templates as one typically wants *u*nivariates before *b*ivariates.

na_first_in_section

Whether to place NAs first when sorting

scalar<logical> // *default:* TRUE (optional)

Default ascending and descending sorting with dplyr::arrange() is to place NAs at the end. This would have placed univariates at the end, etc. Thus, saros places NAs first in the section. Set this to FALSE to override.

max_width_obj, max_width_chunk, max_width_file

Maximum object width

scalar<integer> // *default:* NULL (optional)

Maximum width for names of objects (in R/Python environment), chunks (#| label:) and optional files. Note, will always replace variable labels with variable names, to avoid very long file names. Note for filenames: Due to OneDrive having a max path of about 400 characters, this can quickly be exceeded with a long

path base path, long file names if using labels as part of structure, and hashing with Quarto's `cache: true` feature. Thus consider restricting max_width_file to lower than what you optimally would have wished for.

`max_width_folder_name`

Maximum clean folder name length
`scalar<integer>` // *default:* `NULL` (`optional`)

Whereas `max_width_file` truncates the file name, this argument truncates the folder name. It will not impact the report or chapter names in website, only the folders.

`sep_obj`, `sep_chunk`, `sep_file`

Separator string
`scalar<character>` // *default:* `"_"` (`optional`)

Separator to use between grouping variables. Defaults to underscore for object names and hyphen for chunk labels and file names.

`filename_prefix`

Prefix string for all qmd filenames
`scalar<character>` // *default:* `""` (`optional`)

For mesos setup it might be useful to set these files (and related sub-folders) with an underscore (`filename_prefix = "_"`) in front as other stub files will include these main qmd files.

`...` *Dynamic dots*

<dynamic-dots>

Arguments forwarded to the corresponding functions that create the elements.

`progress` *Whether to display progress message*
`scalar<logical>` // *default:* `TRUE`

Mostly useful when hide_bi_entry_if_sig_above < 1

`variable_group_dep`

Name for the variable_group_dep column
`scalar<string>` // *default:* `".variable_group_dep"`

This column is used to group variables that are part of the same bivariate analysis.

`variable_group_prefix`

Set a prefix to more easily find it in your labels
`scalar<string>` // *default:* `NULL`

By default, the .variable_group column is just integers. If you wish to use this as part of your object/label/filename numbering scheme, a number by itself will not be very informative. Hence you could set a prefix such as "Group" to distinguish this column from other columns in the chapter_structure.

`n_range_glue_template_1`, `n_range_glue_template_2`

`scalar<string>` // *default:* `"{n}" and "[{n[1]}, {n[2]}]` (`optional`)

Glue templates for the n_range columns to be created.

`log_file` *Path to log file*
`scalar<string>` // *default:* `"_log.txt"` (`optional`)

Path to log file. Set to NULL to disable logging.

Value

A grouped tibble (data.frame) with columns that fall into two main categories:

Input columns (from user data):

- `chapter` (character): Chapter name (input)
- `dep` (character): Dependent variable selector (input)
- `indep` (character, optional): Independent variable selector (input)

Constructed columns (all start with a dot):

- `.variable_name`, `.variable_position` (character/integer): Variable name and position
- `.variable_label`, `.variable_label_prefix`, `.variable_label_suffix` (character): Variable label and its components
- `.variable_type`, `.variable_type_dep`, `.variable_type_indep` (character): Variable type(s)
- `.variable_name_dep`, `.variable_name_indep` (character): Names of dependent/independent variables
- `.variable_label_prefix_dep`, `.variable_label_prefix_indep` (character): Label prefixes for dep/indep
- `.variable_group_dep` (character/factor): Grouping variable for bivariate analysis
- `.variable_group_id` (integer): Numeric group identifier for bivariate analysis
- `.chapter_number` (integer): Chapter number
- `.template_name` (character): Name of chunk template used
- `.obj_name`, `.chunk_name`, `.file_name` (character): Object, chunk, and file names (for output)
- `.n`, `.n_range` (integer/character): Sample size and range
- `.n_cats_dep`, `.n_cats_indep` (integer): Number of categories for dep/indep
- `.max_chars_labels_dep`, `.max_chars_labels_indep` (integer): Max label length for dep/indep
- `.max_chars_cats_dep`, `.max_chars_cats_indep` (integer): Max category label length for dep/indep
- `.n_dep`, `.n_indep` (integer): Number of dep/indep variables in group
- `.bi_test`, `.p_value` (character/numeric): Statistical test name and p-value for bivariates
- `.keep_bi_rows` (logical): Whether bivariate row is kept
- Other columns may be present depending on chunk templates and options.

Row count estimate:

- The number of rows in the output depends on the number of chapters, dep/indep combinations, and chunk templates. Typically, it is the sum of all unique variable combinations specified in `chapter_overview`, expanded by chunk templates and filtered by significance and other options. For a simple overview, expect one row per variable per chapter; for bivariates, one row per dep-indep pair.

Grouping variables:

- The columns used for grouping (i.e., `dplyr::grouped_df`) are determined by the `organize_by` argument. By default, this includes `.chapter_number`, `.variable_label_prefix_dep`, `.variable_name_indep`, and `.template_name`, but can be customized. These columns define how the output is grouped for further analysis or reporting.

See function source and documentation for details on each column's meaning and usage.

Examples

```
ref_df <- refine_chapter_overview(
  chapter_overview = ex_survey_ch_overview
)
```

`remove_entry_from_sidebar` *Removes entries in sidebar if containing a filename regex pattern.*

Description

Removes entries in sidebar if containing a filename regex pattern.

Usage

```
remove_entry_from_sidebar(
  path = "_site",
  filename_as_regex = c("report\\.pdf", "report\\.docx")
)
```

Arguments

`path`	String, path to where your html-files are located. Defaults to "_site"
`filename_as_regex`	Character vector of regex patterns to search for. Defaults to c("report\.pdf", "report\.docx")

Value

Invisibly returns files processed

sanitize_chr_vec *Sanitize character vector, for instance useful for variable label in* `labelled::update_variable_labels_with()`

Description

Sanitize character vector, for instance useful for variable label in `labelled::update_variable_labels_with()`

Usage

```
sanitize_chr_vec(
  x,
  sep = " - ",
  multi_sep_replacement = ": ",
  replace_ascii_with_utf = FALSE
)
```

Arguments

x	Character vector or factor vector
sep	String, separates prefix (e.g. main question) from suffix (sub-question)
multi_sep_replacement	String. If multiple separators (sep) are found, replace the first ones with this.
replace_ascii_with_utf	Flag. If TRUE, converts HTML characters to Unicode symbol.

Value

Character vector with sanitized strings

Examples

```
# Example 1: Basic usage
input <- c("<b>Bold</b>", "  Extra spaces   ", "- Selected Choice -")
sanitize_chr_vec(input)

# Example 2: Replace ASCII with UTF
input <- c("&Agrave;", "&Eacute;", "&Ouml;")
sanitize_chr_vec(input, replace_ascii_with_utf = TRUE)

# Example 3: Custom separators
input <- c("Question - Subquestion", "Another - Example")
sanitize_chr_vec(input, sep = " - ", multi_sep_replacement = ": ")
```

setup_access_restrictions *Setup files needed for basic password-based access restriction for website*

Description

Create a _headers file for 'Netlify' publishing or a set of .htaccess and .htpasswd files (FTP) placed in the specific subfolders.

Usage

```
setup_access_restrictions(
  remote_basepath = "/home/",
  local_basepath,
 rel_path_base_to_parent_of_user_restricted_folder = file.path("Reports", "2022",
    "Mesos"),
  warn = TRUE,
  local_main_password_path = ".main_htpasswd_public",
  username_folder_matching_df = NULL,
  universal_usernames = c("admin"),
  log_rounds = 12,
  append_users = TRUE,
  password_input = "prompt",
  type = c("netlify", "apache"),
  create_main_htaccess = FALSE
)
```

Arguments

remote_basepath	String. Folder where site will be located if using FTP-server. Needed for .htaccess-files.
local_basepath	String. Local folder for website, typically "_site".
rel_path_base_to_parent_of_user_restricted_folder	String, relative path from basepath to the folder where the restricted folders are located. (E.g. the "mesos"-folder)
warn	Flag. Whether to provide warning or error if paths do not exist.
local_main_password_path	String. Path to main file containing all usernames and passwords formatted with a colon between username and password.
username_folder_matching_df	Data frame. If NULL (default), will use folder names as usernames. Otherwise, a data frame with two columns: "folder" and "username" where "folder" is the name of the folder and "username" is the username for that folder.

universal_usernames
: Character vector. Usernames in local_main_htpasswd_path which always have access to folder

log_rounds
: Integer, number of rounds in the bcrypt algorithm. The higher the more time consuming and harder to brute-force.

append_users
: Boolean, if TRUE (default) will create new users and add them to local_main_password_path. See also password_input.

password_input
: String, either "prompt" which asks the user for input. Alternatively, a number stored as string for a generated random password of said length: "8", "10", "12", "16"

type
: Character vector. "netlify" will create _headers file used for Netlify. "apache" will create .htaccess and .htpasswd files used for general FTP-servers.

create_main_htaccess
: Logical. If TRUE, creates a main .htaccess file in local_basepath with security headers (HSTS, X-Content-Type-Options, X-Frame-Options, etc.). Only applicable when type includes "apache". Default is FALSE.

Value

String, the path to the newly created _headers-file or .htaccess files.

`setup_mesos` *Simply create qmd-files and yml-files for mesos reports*

Description

Simply create qmd-files and yml-files for mesos reports

Usage

```
setup_mesos(
  main_directory = character(),
  mesos_var_subfolder = character(),
  files_to_process,
  mesos_df,
  files_taking_title = c("index.qmd", "report.qmd"),
  read_syntax_pattern = "qs::qread\\('",
  read_syntax_replacement = "qs::qread('../../",
  qmd_regex = "\\.qmd",
  subtitle_separator = " - ",
  prefix = "{{< include \"",
  suffix = "\" >}}"
)
```

Arguments

`main_directory`	String, path to where the _metadata.yml, stub QMD-files and their subfolders are created.
`mesos_var_subfolder`	String, optional name of a subfolder of the mesos_var folder in where to place all mesos_group folders.
`files_to_process`	Character vector of files used as templates for the mesos stubs.
`mesos_df`	List of single-column data frames where each variable is a mesos variable, optionally with a variable label indicating its pretty name. The values in each variable are the mesos groups. NA is silently ignored.
`files_taking_title`	Character vector of files for which titles should be set. Optional but recommended.
`read_syntax_pattern, read_syntax_replacement`	Optional strings, any regex pattern to search and replace in the qmd-files. If NULL, will ignore it.
`qmd_regex`	String. Experimental feature for allowing Rmarkdown, not yet tested.
`subtitle_separator`	String or NULL. If a string will add title and subtitle fields to the _metadata.yml-files in the deepest child folders. The title is the mesos_group. The subtitle is a concatenation of the folder name of the main_directory and the mesos_var label.
`prefix, suffix`	String for the include section of the stub qmd files.

`setup_mesos_structure` *Create directory structure for mesos reports (improved version)*

Description

This is an improved, easier-to-use version of `setup_mesos()`. It creates the directory structure, QMD stub files, and YAML metadata files needed for mesos (multi-group) reports without requiring manual working directory management.

Usage

```
setup_mesos_structure(
  main_directory,
  files_to_process,
  mesos_groups,
  mesos_var_subfolder = character(),
  files_taking_title = c("index.qmd", "report.qmd"),
  subtitle_separator = " - ",
  include_prefix = "{{< include \"",
  include_suffix = "\" >}}"
)
```

Arguments

`main_directory` String. Path to where the structure will be created. Can be an absolute path or relative path. The path will be created if it doesn't exist. Unlike `setup_mesos()`, this parameter is required and has no default to avoid accidental file creation in unexpected locations.

`files_to_process`
Character vector of paths to template QMD files to use as the basis for creating stub files. These files should typically have filenames starting with underscore (e.g., `_report.qmd`).

`mesos_groups` A named list or data frame specifying the grouping structure.

- If a **named list**: names are mesos variable names, values are character vectors of group names. Example: `list(region = c("North", "South", "East"))`
- If a **data frame**: Use the same format as `setup_mesos()` - a list of single-column data frames with optional variable labels.

`mesos_var_subfolder`
Optional character vector. Subfolder path(s) within each mesos variable folder where group folders should be placed. Default is no subfolder (empty character vector).

`files_taking_title`
Character vector of filenames that should receive title metadata. Default is `c("index.qmd", "report.qmd")`.

`subtitle_separator`
String or NULL. If a string, adds title and subtitle fields to `_metadata.yml` files in the deepest child folders. The subtitle is a concatenation of the output directory basename, mesos variable label, and group name. Default is `" - "`. Set to `NULL` to disable.

`include_prefix, include_suffix`
Strings for the include directive in stub QMD files. Default creates Quarto-style includes: `{{< include "..." >}}`

Value

Invisibly returns a list with information about created files.

Examples

```
## Not run:
# Simple example with a named list
setup_mesos_structure(
  main_directory = "reports/2024",
  files_to_process = c("_report.qmd", "_index.qmd"),
  mesos_groups = list(
    region = c("North", "South", "East", "West"),
    department = c("Sales", "Marketing", "IT")
  )
)
```

```
# With subfolder and custom labels
setup_mesos_structure(
  main_directory = "./output",
  files_to_process = "_analysis.qmd",
  mesos_groups = list(
    country = c("Norway", "Sweden", "Denmark")
  ),
  mesos_var_subfolder = "reports/Q1"
)

## End(Not run)
```

`write_default_draft_report_args` *Write Default Arguments for* `draft_report()` *to YAML-file*

Description

Write Default Arguments for `draft_report()` to YAML-file

Usage

```
write_default_draft_report_args(
  path,
  ignore_args = c("data", "...", "dep", "indep", "chapter_structure", "chapter_overview",
    "path")
)
```

Arguments

`path`	`scalar<character>` // Required. *default:* `settings.yaml`
`ignore_args`	`vector<character>` // Optional. *default:* `c("data", "...", "dep", "indep", "chapter_structure", "chapter_overview")` A character vector of argument (names) not to be written to file.

Value

The defaults as a `yaml`-object.

Examples

```
write_default_draft_report_args(path = tempfile(fileext = ".yaml"))
```

Index