

# Package ‘saros’

February 29, 2024

**Type** Package

**Title** Semi-Automatic Reporting of Ordinary Surveys

**Version** 1.0.4

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

**Description** Produces highly customized reports, primarily intended for survey research. Building on 'Quarto' (<<https://quarto.org>>), it generates draft chapters of all specified dependent/independent variables, which can be further edited by hand, containing figures, tables and analyses (currently only uni-/bivariate tests of equivalent means/proportions). The feature 'mesos'-reports offer tailor-made batch report production where e.g. an institution can compare itself to all other participants. Publication tools are also included, such as password generation for 'mesos'-report sections on a 'Quarto' Website.

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

**License** MIT + file LICENSE

**URL** <https://nifu-no.github.io/saros/>, <https://github.com/NIFU-NO/saros>

**BugReports** <https://github.com/NIFU-NO/saros/issues>

**Depends** R (>= 4.2.0)

**Imports** cli, bcrypt, utils, vctrs, dplyr, tidyr, tidyselect, glue, rlang, stringi, yaml, forcats, ggplot2, ggiraph, mschart, officer, fs, clipr, rstudioapi, rvest

**Suggests** covr, srvyr, writexl, haven, readr, ggfittext, labelled, testthat (>= 3.0.0), readxl, htmltools, reactable, stringr, tibble, qs, openai, withr, quarto, spelling

**SystemRequirements** Quarto command line tools (<https://github.com/quarto-dev/quarto-cli>) for the rendering wrapper functionality.

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.1

**Language** en-US

**Config/Needs/website** rmarkdown

**Config/testthat/parallel** true

**Config/testthat/start-first** refine\_chapter\_overview

**NeedsCompilation** no

**Author** Stephan Daus [aut, cre, cph] (<<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** 2024-02-29 01:10:02 UTC

## R topics documented:

attach_qualtrics_labels . . . . .	3
center_string . . . . .	4
col_to_binaries . . . . .	4
combn_upto . . . . .	5
create_caption . . . . .	6
create_email_credentials . . . . .	7
crostable3 . . . . .	8
draft_report . . . . .	8
embed_cat_freq_plot . . . . .	20
embed_cat_freq_plot_docx . . . . .	21
embed_cat_prop_plot . . . . .	23
embed_cat_prop_plot_docx . . . . .	24
embed_cat_table . . . . .	26
embed_cat_text_html . . . . .	27
embed_chr_table_html . . . . .	28
embed_sigtest . . . . .	29
ex_survey . . . . .	30
ex_survey_ch_overview . . . . .	31
filename_sanitizer . . . . .	32
get_colour_palette . . . . .	32
get_colour_set . . . . .	33
get_data_label_opts . . . . .	35
handpick . . . . .	35
hex_bw . . . . .	36
is_colour . . . . .	36
list_available_element_types . . . . .	37
omitted_recoder_df . . . . .	37
post_render_docx_img_replacer . . . . .	39
prepare_chunk . . . . .	40
prepare_safe_render . . . . .	41

quarto_render_saros . . . . .	42
read_default_draft_report_args . . . . .	42
refine_chapter_overview . . . . .	43
remove_label_parts . . . . .	44
remove_special_chars_in_labels . . . . .	45
rename_by_labels . . . . .	45
render_full_reports . . . . .	46
replace_docx_imgs_with_mscharts . . . . .	47
replace_stata_labels . . . . .	48
sanitize_labels . . . . .	48
setup_access_restrictions . . . . .	49
subset_vector . . . . .	51
summarize_data . . . . .	51
swap_label_colnames . . . . .	52
write_default_draft_report_args . . . . .	53

**Index****54**

attach\_qualtrics\_labels

*Re-attach label information from Qualtrics not obtained from regular data downloads*

**Description**

Re-attach label information from Qualtrics not obtained from regular data downloads

**Usage**

```
attach_qualtrics_labels(
  data,
  questions,
  reverse_stata_replacement = FALSE,
  questions_var = "qname",
  questions_question = "question"
)
```

**Arguments**

data	Data.frame with original variable names.
questions	Data frame with questions obtained from <code>qualtRics::survey_questions()</code>
reverse_stata_replacement	If variable names have already been modified
questions_var	String, indicating column name in questions that indicates column names.
questions_question	String, indicating column name in questions for the full question. with full stops changed to underscores, this will reverse them for connection. Rarely needed. Defaults to FALSE.

**Value**

Data returned with only variable labels modified.

---

center_string	<i>Center String Vector</i>
---------------	-----------------------------

---

**Description**

Center String Vector

**Usage**

```
center_string(string, maxwidth = 50)
```

**Arguments**

string	String vector
maxwidth	Maximum width

**Value**

String vector

**Examples**

```
center_string(string=c("This is a very long label for a graph.",
  "But this one is even longer due to superfluous and verbose way of writing"),
  maxwidth=20)
```

---

col_to_binaries	<i>Mutate a (factor, character, integer, etc) column into multiple columns,</i>
-----------------	---

---

**Description**

Easily mutate a single column into multiple columns (~dummies+1), while retaining variable labels and order of the original factor variable.

**Usage**

```
col_to_binaries(data, col, var_separator = "___", label_separator = " - ")
```

**Arguments**

data	<i>Survey data</i> obj:<data.frame> obj:<tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
col	Single column. Tidy-select.
var_separator	<i>Variable separator</i> scalar<character> // default: NULL (optional) Separator between old variable name and categories.
label_separator	scalar<character> // default: NULL (optional) Separator between old label name and new label part.

**Value**

Original data frame with the binary columns attached, containing new labels.

**Examples**

```
col_to_binaries(ex_survey, col = b_3, label_separator = " - ")
```

---

combn_upto	<i>Create All Possible Combinations of Vector Elements with Minimum A and Maximum B.</i>
------------	--

---

**Description**

Create All Possible Combinations of Vector Elements with Minimum A and Maximum B.

**Usage**

```
combn_upto(
  vec = c("a", "b", "c", "d", "e", "f", "g"),
  n_min = 6L,
  n_max = length(vec)
)
```

**Arguments**

vec	Vector
n_min	Minimum number of elements
n_max	Maximum number of elements. Defaults to length of vec.

**Value**

A data frame

**Examples**

```
combn_upto()
```

---

create_caption	<i>Adds caption attribute</i>
----------------	-------------------------------

---

**Description**

Adds caption attribute

**Usage**

```
create_caption(
  main_question,
  data_out,
  indep_pos = NULL,
  mesos_group = NULL,
  filepath = NULL,
  translations = NULL
)
```

**Arguments**

main_question	String from get_main_question2
data_out	Output from summarize_data
indep_pos	Named integer for the by-variable.
mesos_group	String, indicating the name of the mesos group
filepath	String, path to pretty tabular file
translations	List of named strings for by and (N=, etc. see getOption("saros")\$translations.

**Value**

String

---

 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(
  local_basepath = getwd(),
  rel_path_base_to_parent_of_user_restricted_folder,
  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

`local_basepath` String. Path to where your QMD-files are located (the site to be rendered).

`rel_path_base_to_parent_of_user_restricted_folder` String. Path going from basepath to the folder containing folders to password-protect.

`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

---

crosstable3	<i>Internal function for fast cross-table</i>
-------------	---

---

**Description**

Internal function for fast cross-table

**Usage**

```
crosstable3(data, ...)
```

**Arguments**

data	data.frame, survey object (requires srvyr-package)
...	Dynamic dots

**Value**

Data.frame

---

draft_report	<i>Automatically Draft a Quarto Report</i>
--------------	--

---

**Description**

The `draft_report()` function is the main function, and the only necessary user interface, to create semi-automated (draft) reports. It does not need to be the first step, however, as one might want to store and read in arguments for the function with the `read_yaml_params()`-function first. After the report files has been drafted with `draft_report()`, you can edit, render, and ultimately publish these as usual with Quarto features in RStudio. The `index.qmd` will be the main output file containing "includes" to other chapters.

**Usage**

```
draft_report(
  data,
  chapter_overview = NULL,
  ...,
  path,
  title = "Report",
  authors = NULL,
  mesos_report = FALSE,
```



```
mesos_var = NULL,
label_separator = " - ",
name_separator = NULL,
index_yaml_file = NULL,
report_yaml_file = NULL,
chapter_yaml_file = NULL,
qmd_start_section_filepath = NULL,
qmd_end_section_filepath = NULL,
index_filename = "index.qmd",
element_names = c("uni_cat_prop_plot", "uni_cat_freq_plot", "uni_cat_table",
  "uni_chr_table", "hline", "bi_catcat_prop_plot", "bi_catcat_freq_plot",
  "bi_catcat_prop_plot2", "bi_catcat_freq_plot2", "bi_catcat_table", "bi_sigtest"),
sort_by = ".upper",
data_label = saros::get_data_label_opts(),
always_show_bi_for_indep = NULL,
categories_treated_as_na = NULL,
variables_always_at_top = NULL,
variables_always_at_bottom = NULL,
return_raw = TRUE,
showNA = c("never", "always", "ifany"),
totals = FALSE,
hide_bi_entry_if_sig_above = 1,
hide_test_if_n_below = 10,
hide_result_if_n_below = 10,
hide_chr_for_others = TRUE,
hide_variable_if_all_na = TRUE,
single_y_bivariates_if_indep_cats_above = 3,
single_y_bivariates_if_deps_above = 20,
digits = 1,
data_label_decimal_symbol = ".",
hide_label_if_prop_below = 0.01,
hide_axis_text_if_single_variable = FALSE,
main_font_size = 10,
label_font_size = 3,
strip_font_size = 7,
legend_font_size = 7,
strip_width = 15,
strip_angle = 0,
x_axis_label_width = 20,
plot_height_multiplier_per_horizontal_line = NA,
plot_height_multiplier_per_vertical_letter = 0.2,
plot_height_multiplier_per_facet = 0.95,
plot_height_multiplier_per_legend_line = 1.1,
plot_height_fixed_constant = 0,
plot_height_max = 8,
plot_height_min = 1.5,
vertical_height = 12,
vertical = FALSE,
```

```

png_scale = 1.2,
png_width = 14,
png_height = 16,
font_family = "sans",
colour_palette_nominal = NULL,
colour_palette_ordinal = NULL,
colour_na = "gray90",
colour_2nd_binary_cat = NULL,
table_main_question_as_header = FALSE,
max_width_obj = 128,
max_width_file = 64,
max_clean_folder_name = 12,
open_after_drafting = FALSE,
organize_by = c("chapter", ".variable_label_prefix_dep", ".variable_name_indep",
  ".element_name"),
arrange_output_by = c("chapter", ".variable_name_dep", ".variable_name_indep"),
ignore_heading_for_group = c(".element_name", ".variable_type_dep", "chapter"),
replace_heading_for_group = c(.variable_label_suffix_dep = ".variable_name_dep"),
mesos_first = TRUE,
descend = TRUE,
require_common_categories = TRUE,
panel_tabset_mesos = TRUE,
pdf = TRUE,
attach_chapter_dataset = TRUE,
auxiliary_variables = NULL,
flexi = FALSE,
micro = FALSE,
reps = 1000,
information = c(".variable_label_dep", ".category", ".count", ".count_se",
  ".proportion", ".proportion_se", ".mean", ".mean_se", ".data_label",
  ".comb_categories", ".sum_value"),
contents = c("intro", "not_used_category", "mode_max", "value_max", "value_min",
  "value_diff", "mean_max", "mean_min", "mean_diff", "median_max", "median_min",
  "median_diff", "variance_max", "variance_min"),
include_numbers = TRUE,
n_top_bottom = 1,
log_file = NULL,
serialized_format = c("rds", "qs"),
tabular_format = c("delim", "xlsx", "csv", "csv2", "tsv", "sav", "dta"),
translations = list(last_sep = " and ", download_report = "Download report (PDF)",
  intro_prefix = "We will now look at the questions asked regarding ", intro_suffix =
  "", mode_max_onfix = " on ", mode_max_prefix = "The most common responses were ",
  mode_max_suffix = "", not_used_prefix =
  "The following response categories were not used: ", not_used_suffix = "",
  value_max_prefix = "", value_max_infix =
  " {?is/are} the {dots$n_top_bottom} item{?s} where the most responded ",
  value_max_suffix = "", value_min_prefix = "",
  value_min_infix =

```

```

" {?is/are} the {dots$n_top_bottom} item{?s} where the fewest responded ",
value_min_suffix = "", mean_onfix = "M = ", mean_max_prefix =
"They have highest mean on ", mean_max_suffix = "", mean_min_prefix =
"They have lowest mean on ", mean_min_suffix = "", median_onfix = "Median = ",
median_max_prefix = "They have highest median on ", median_max_suffix = "",
median_min_prefix = "They have lowest median on ", median_min_suffix = "",
intro_by_prefix = "We will now look at the questions asked regarding ",

intro_by_infix = " broken down by ", intro_by_suffix = "", by_breakdown = " by ",
n_equal_prefix = " (N &equals; ", n_equal_suffix = ") ", table_heading_N =
"Total (N)", by_total = "Everyone", sigttest_prefix = "Significance testing of ",
sigttest_suffix = "", mesos_group_prefix = " Group: ", mesos_group_suffix = "",
mesos_label_all_others = "Others", empty_chunk_text = "\nText\n",
flexi_input_chapter = "Chapter(s):", flexi_input_dep = "Dependent variable(s):",
flexi_input_indep = "Independent variable:",
flexi_input_mesos_group =
"Filter:", flexi_figure_type = "Figure type:", flexi_data_label =
"Summary to display", flexi_showNA = "Show NA (Missing)", flexi_sort_by = "Sort by",
flexi_totals = "Totals", flexi_digits = "Digits after decimal", flexi_table =
"Table", flexi_figure = "Figure", flexi_cols_variable_name = "Variable name",
flexi_cols_variable_label = "Variable label", flexi_cols_category =
"Response category", flexi_cols_count = "N", flexi_cols_count_se = "SE(N)",
flexi_cols_proportion = "Proportion",
flexi_cols_proportion_se =
"SE(Proportion)", flexi_cols_mean = "Mean", flexi_cols_mean_se = "SE(Mean)",
flexi_cols_data_label = "Data label", flexi_cols_comb_categories =
"Combined categories", flexi_cols_sum_value =
"Sum of data label across combined categories", flexi_validate =
"Error: Columns must have some categories in common.", flexi_settings = "Settings",
flexi_basic_settings = "Basic", flexi_advanced_settings = "Advanced",
flexi_input_indep_none = "<none>", flexi_figure_type_proportion = "Proportion",

flexi_figure_type_frequency = "Frequency", flexi_hide_label_if_prop_below =
"Hide label if proportion below:")
)

```

## Arguments

**data** *Survey data*  
obj: <data.frame>|obj: <tbl\_df> // Required  
A data frame (or a srvyr-object) with the columns specified in the chapter\_overview 'dep\_cat', etc columns.

**chapter\_overview** *What goes in each 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.

...	<i>Dynamic dots</i> <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.
path	<i>Output path</i> scalar<character> // default: NULL (optional) Path to save all output.
title	<i>Title of report</i> scalar<character> // default: NULL (optional) Added automatically to YAML-header of index.qmd-file.
authors	<i>Authors of entire report</i> vector<character> // default: NULL (optional) If NULL, infers from chapter_overview\$authors, and collates for entire report.
mesos_report	<i>Whether to produce reports per mesos group</i> scalar<logical> // default: FALSE If false, returns a regular single report.
mesos_var	<i>Variable in 'data' indicating groups to tailor reports for</i> scalar<character> // default: NULL (optional) Column name in data indicating the groups for which mesos reports will be produced.
label_separator	<i>Variable label separator</i> scalar<character> // default: NULL (optional) String to split labels on main question and sub-items.
name_separator	<i>Variable name separator</i> scalar<character> // default: NULL (optional) String to split column names in data between main question and sub-items
index_yaml_file, report_yaml_file	<i>Path to YAML-file to insert into index.qmd and report.qmd respectively</i> scalar<character> // default: NULL (optional) Path to file used to insert header YAML, in index and report files.
chapter_yaml_file	<i>Path to YAML-file to insert into each chapter qmd-file</i> scalar<character> // default: NULL (optional) Path to file used to insert header YAML, in each chapter.
qmd_start_section_filepath, qmd_end_section_filepath	<i>Path to qmd-bit for start/end of each qmd</i> scalar<character> // default: NULL (optional) Path to qmd-snippet placed before/after body of all chapter qmds.
index_filename	<i>Index filename</i> scalar<character> // default: "index.qmd" (optional) The name of the main index Quarto file (and its subfolder) used as landing page for each report. Will link to a PDF (report.qmd) which collects all chapters.
element_names	<i>Elements to be reported</i> vector<character> // default: NULL (optional) Elements to be reported for all sets (batteries) of y-variables.

sort_by	<p><i>What to sort output by</i></p> <p>vector&lt;character&gt; // default: NULL (optional)</p> <p>Sort output (and collapse if requested).</p> <p><b>".top"</b> The proportion for the highest category available in the variable.</p> <p><b>".upper"</b> The sum of the proportions for the categories above the middle category.</p> <p><b>".mid_upper"</b> The sum of the proportions for the categories including and above the middle category.</p> <p><b>".mid_lower"</b> The sum of the proportions for the categories including and below the middle category.</p> <p><b>".lower"</b> The sum of the proportions for the categories below the middle category.</p> <p><b>".bottom"</b> The proportions for the lowest category available in the variable.</p> <p><b>".variable_label"</b> Sort by the variable labels.</p> <p><b>".id"</b> Sort by the variable names.</p> <p><b>".by_group"</b> The groups of the by argument.</p> <p><b>character()</b> Character vector of category labels to sum together.</p>
data_label	<p><i>Data label</i></p> <p>scalar&lt;character&gt; // default: "proportion" (optional)</p> <p>One of "proportion", "percentage", "percentage_bare", "count", "mean", or "median".</p>
always_show_bi_for_indep	<p><i>Always show bivariate for indep-variable</i></p> <p>vector&lt;character&gt; // default: NULL (optional)</p> <p>Specific combinations with a by-variable where bivariates should always be shown.</p>
categories_treated_as_na	<p><i>NA categories</i></p> <p>vector&lt;character&gt; // default: NULL (optional)</p> <p>Categories that should be treated as NA.</p>
variables_always_at_top, variables_always_at_bottom	<p><i>Top/bottom variables</i></p> <p>vector&lt;character&gt; // default: NULL (optional)</p> <p>Column names in data that should always be placed at the top or bottom of figures/tables.</p>
return_raw	<p><i>NOT IN USE</i></p> <p>scalar&lt;integer&gt; // default: FALSE</p> <p>Whether to return the raw static element.</p>
showNA	<p><i>Show/hide NA in categorical variables</i></p> <p>scalar&lt;logical&gt; // default: NULL (optional)</p> <p>Whether to show NA in categorical variables (one of c("ifany", "always", "never")).</p>
totals	<p><i>Include totals</i></p> <p>scalar&lt;logical&gt; // default: FALSE (optional)</p> <p>Whether to include totals in the output.</p>

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\_test\_if\_n\_below  
*Threshold n for hiding significance test*  
 scalar<integer> // default: 0 (optional)  
 If N is below this value, p-value will not be shown.

hide\_result\_if\_n\_below  
*Hide result if N below*  
 scalar<integer> // default: 10 (optional)  
 Whether to hide result if N for a given datasets (or mesos group) 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\_chr\_for\_others  
*Hide open response displays for others*  
 scalar<logical> // default: TRUE (optional)  
 For mesos reports using the element "chr\_table", open responses are displayed for also the entire sample (FALSE) or only for the mesos group to ensure data privacy (TRUE).

hide\_variable\_if\_all\_na  
*Hide variable from outputs if containing all NA*  
 scalar<boolean> // default: TRUE (optional)  
 Whether to remove all variables (in particular useful for mesos) if all values are NA

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.

digits  
*Decimal places*  
 scalar<integer> // default: 0L (optional)  
 Number of decimal places.

data\_label\_decimal\_symbol  
*Decimal symbol*  
 scalar<character> // default: "." (optional)

Decimal marker, some might prefer a comma ',' or something else entirely.  
NOTE: Future version will likely postpone formatting this until `gt()`, `kable()`, etc.

`hide_label_if_prop_below`  
*Hide label threshold*  
scalar<numeric> // default: NULL (optional)  
Whether to hide label if below this value. NOTE: Future versions will likely distinguish between `element_types`.

`hide_axis_text_if_single_variable`  
*Hide y-axis text if just a single variable*  
scalar<boolean> // default: FALSE (optional)  
Whether to hide text on the y-axis label if just a single variable

`main_font_size, label_font_size, strip_font_size, legend_font_size`  
*Font sizes*  
scalar<integer> // default: 12 (optional)  
Font sizes for general text (10), data label text (3), strip text (7) and legend text (7).

`strip_angle` *Angle on the facet strip in plots*  
scalar<double> // default: 0

`x_axis_label_width, strip_width`  
*Label width of x-axis and strip texts in plots*  
scalar<integer> // default: 20 (optional)  
Width of the labels used for the categorical column names in x-axis texts and strip texts.

`plot_height_multiplier_per_vertical_letter, plot_height_multiplier_per_horizontal_line`  
*Height multiplier*  
scalar<double> // default: .1  
Height in cm per chart entry, for all static plots.

`plot_height_multiplier_per_facet`  
*Plot height multiplier per facet*  
scalar<double> // default: 0.95 (optional)  
Multiplier for plot height per facet. Defaults to optimal at .95, i.e. slightly less than no change (1).

`plot_height_multiplier_per_legend_line`  
*Plot height multiplier per legend line*  
scalar<double> // default: 1.1 (optional)  
Multiplier for plot height per horizontal line of legend. Defaults to optimal at 1.1, i.e. slightly more than no change (1).

`plot_height_fixed_constant`  
*Height constant addition*  
scalar<double> // default: 0  
Fixed height in cm to add to all static plots.

`plot_height_max`  
*Maximum plot height*  
scalar<double> // default: 10 (optional)  
Maximum height for the plot.

plot_height_min	<i>Minimum plot height</i> scalar<double> // default: 2 (optional) Minimum height for the plot.
vertical_height	<i>Vertical height</i> scalar<double> // default: NULL (optional) Height for vertical layout of plot? NEEDS CHECKING
vertical	<i>Orientation of plots</i> scalar<logical> // default: FALSE (optional) If FALSE (default), then horizontal plots.
png_scale	<i>PNG scale</i> scalar<double> // default: 1 (optional) Scale factor for PNG output.
png_width, png_height	<i>PNG width and height</i> scalar<double> // default: 12 (optional) Width for PNG output.
font_family	<i>Font family</i> scalar<character> // default: "sans" (optional) Word font family. See officer::fp_text.
colour_palette_nominal, colour_palette_ordinal	<i>Colour palettes (nominal and ordinal)</i> vector<character> // default: NULL (optional) Must contain at least the number of unique values (including missing) in the data set.
colour_na	<i>Colour for NA category</i> scalar<character> // default: NULL (optional) Colour as a single string for NA values.
colour_2nd_binary_cat	<i>Colour for second binary category</i> scalar<character> // default: "#ffffff" (optional) Colour for second category in binary variables. Often useful to hide this.
table_main_question_as_header	<i>Table main question as header</i> scalar<logical> // default: FALSE (optional) Whether to include the main question as a header in the table.
max_width_obj	<i>Maximum object width</i> scalar<integer> // default: NULL (optional) Maximum width for object names in the Quarto script. In particular useful when having label as part of the structure.
max_width_file	<i>Maximum filename width</i> scalar<integer> // default: NULL (optional) Maximum width for any filename. 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. This argument truncates the filenames.

max_clean_folder_name	<p><i>Maximum clean folder name length</i>          scalar&lt;integer&gt; // 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.</p>
open_after_drafting	<p><i>Whether to open index.qmd</i>          scalar&lt;logical&gt; // default: FALSE (optional)          Whether to open the main output file (index.qmd) after completion.</p>
organize_by	<p><i>Grouping columns</i>          vector&lt;character&gt; // default: NULL (optional)          Column names used for identifying chapters and sections.</p>
arrange_output_by	<p><i>Grouping columns</i>          vector&lt;character&gt; // default: NULL (optional)          Column names used for sorting output within each organize_by group</p>
ignore_heading_for_group	<p><i>Ignore heading for group</i>          vector&lt;character&gt; // default: NULL (optional)          Type of refined chapter_overview data for which to suppress the heading in the report output. Typically variable_name_dep, variable_name_indep, etc.</p>
replace_heading_for_group	<p><i>Replacing heading for group</i>          named vector&lt;character&gt; // 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_overview. 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.</p>
mesos_first	<p><i>mesos first</i>          scalar&lt;logical&gt; // default: FALSE (optional)          Whether to place the mesos group element before or after the entire sample.</p>
descend	<p><i>Sorting order</i>          scalar&lt;logical&gt; // default: FALSE (optional)          Reverse sorting of sort_by.</p>
require_common_categories	<p><i>Check common categories</i>          scalar&lt;logical&gt; // default: NULL (optional)          Whether to check if all items share common categories.</p>
panel_tabset_mesos	<p><i>mesos panel tabset</i>          scalar&lt;logical&gt; // default: TRUE (optional)</p>

	Whether in mesos reports the comparison group should be displayed as a Quarto panel tabset (TRUE), or above each other (FALSE).
pdf	<p><i>Create PDF of full report?</i>          scalar&lt;logical&gt; // default: FALSE (optional)</p> <p>Whether to create a PDF of the entire report (all chapters included in a single file).</p>
attach_chapter_dataset	<p><i>Toggle inclusion of chapter-specific datasets in qmd-files</i>          scalar&lt;logical&gt; // default: FALSE</p> <p>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.</p>
auxiliary_variables	<p><i>Auxiliary variables to be included in datasets</i>          vector&lt;character&gt; // default: NULL (optional)</p> <p>Column names in data that should always be included in datasets for chapter qmd-files, if attach_chapter_dataset=TRUE. Not publicly available.</p>
flexi	<p><i>Create page with user-editable categorical plots and tables</i>          scalar&lt;logical&gt; // default: FALSE (optional)</p> <p>Whether to create a folder with a Shiny flexi app containing all the variables in the chapter_overview and auxiliary_variables.</p>
micro	<p><i>Create page with raw data (micro data) and codebook</i>          scalar&lt;logical&gt; // default: FALSE (optional)</p> <p>Whether to a page with local links to a raw dataset (in various formats) and codebook (in various formats).</p>
reps	<p><i>Number of permutations</i>          scalar&lt;integer&gt; // default: 100 (optional)</p> <p>Number of permutations to be performed in bootstrap significance tests.</p>
information	<p><i>Pre-computed information</i>          scalar&lt;character&gt; // default: NULL (optional)</p> <p>Which pre-computed information for each variable-category to display.</p>
contents	<p><i>Text interpretations</i>          vector&lt;character&gt; // default: all available (optional)</p> <p>The type of text interpretations to return.</p>
include_numbers	<p><i>Include numbers</i>          scalar&lt;logical&gt; // default: NULL (optional)</p> <p>Whether or not to include the actual numbers in parentheses.</p>
n_top_bottom	<p><i>Top and bottom entries to report</i>          scalar&lt;integer&gt; // default: NULL (optional)</p> <p>The number of top and bottom entries to report.</p>
log_file	<p><i>Path to log file</i>          scalar&lt;string&gt; // default: "_log.txt" (optional)</p> <p>Path to log file. Set to NULL to disable logging.</p>

serialized_format	<p><i>Serialized format</i></p> <p>scalar&lt;string&gt; // default: "rds"</p> <p>Format for serialized data. 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.</p>
tabular_format	<p><i>Serialized format</i></p> <p>scalar&lt;string&gt; // default: "delim"</p> <p>Format for pretty tabular data, meant for end-user to peruse and will be linked to in reports (the graph data, etc). One of "delim" (tab-separated delim-files) "xlsx" requires writexl-package), "csv" or "csv2" (requires readr-package. "dta" or "sav" requires haven-package. Currently must be specified, in the future this will become an optional argument.</p>
translations	<p><i>Translations</i></p> <p>list // default: saros::: .saros.env\$defaults\$translations (optional)</p> <p>Named list of strings for translations.</p>

## Details

This function requires at a minimum a dataset (data frame and tibbles are supported so far). Note that saros treats data as they are stored: numeric, integer, factor, ordinal, character, and datetime. Currently, only factor/ordinal and character are implemented. Second, the chapter\_overview must be specified, also as a (small) data frame, with at least the character columns 'chapter' and 'dep', where the first names the output chapters, and the 'dep'-column contain comma-separated (alternatively using tidyselect-syntax) columns in the data which are to be treated as dependent variables. See chapter\_overview for more options.

## Value

Path to index qmd-file. If not specified in the yaml\_path file, will default to index.qmd.

## Examples

```
index_filepath <-
draft_report(
  chapter_overview = ex_survey_ch_overview,
  data = ex_survey,
  path = tempdir())
index_filepaths <-
draft_report(
  chapter_overview = ex_survey_ch_overview,
  data = ex_survey,
  mesos_report = TRUE,
  mesos_var = "f_uni",
  path = tempdir())
```

---

 embed\_cat\_freq\_plot    *Embed Interactive Categorical Plot*


---

## Description

Embed Interactive Categorical Plot

## Usage

```
embed_cat_freq_plot(
  data,
  ...,
  dep = tidyselect::everything(),
  indep = NULL,
  colour_palette = NULL,
  mesos_group = NULL,
  html_interactive = TRUE,
  inverse = FALSE,
  call = rlang::caller_env()
)
```

## Arguments

data	<i>Survey data</i> obj:<data.frame> obj:<tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
...	<i>Dynamic dots</i> <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.
dep, indep	<i>Variable selections</i> <tidyselect> // <i>Default:</i> NULL, meaning everything for dep, nothing for indep. Columns in data. Currently allows tidyselect-syntax, which will be removed.
colour_palette	Character vector of colour codes.
mesos_group	<i>Specific group to compare with</i> scalar<character> // <i>Default:</i> NULL (optional) Both the absolute and relative folderpaths are required, as strings.
html_interactive	<i>Toggle interactive plot</i> scalar<logical> // <i>default:</i> TRUE (optional) Whether plot is to be interactive (ggiraph) or static (ggplot2).
inverse	Flag, defaults to FALSE. If TRUE, swaps x-axis and faceting.
call	<i>Internal call</i> obj:<call> // <i>Default:</i> rlang::caller_env() (optional) Both the absolute and relative folderpaths are required, as strings.

**Value**

ggplot

**Examples**

```
embed_cat_freq_plot(data = ex_survey, dep = b_1:b_3)
```

---

embed\_cat\_freq\_plot\_docx

*Create Word Report with Univariates for Categorical Columns Sharing Same Categories*

---

**Description**

Create Word Report with Univariates for Categorical Columns Sharing Same Categories

Create Word Report with Univariates for Categorical Columns Sharing Same Categories

**Usage**

```
embed_cat_freq_plot_docx(  
  data,  
  ...,  
  inverse = FALSE,  
  dep = tidyselect::everything(),  
  indep = NULL,  
  colour_palette = NULL,  
  mesos_group = NULL,  
  call = rlang::caller_env()  
)
```

```
embed_cat_freq_plot_docx(  
  data,  
  ...,  
  inverse = FALSE,  
  dep = tidyselect::everything(),  
  indep = NULL,  
  colour_palette = NULL,  
  mesos_group = NULL,  
  call = rlang::caller_env()  
)
```

**Arguments**

data	<i>Survey data</i> obj:<data.frame> obj:<tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
...	<i>Dynamic dots</i> <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.
inverse	Flag, defaults to FALSE. If TRUE, swaps x-axis and faceting.
dep, indep	<i>Variable selections</i> <tidyselect> // <i>Default:</i> NULL, meaning everything for dep, nothing for indep. Columns in data. Currently allows tidyselect-syntax, which will be removed.
colour_palette	Character vector of colour codes.
mesos_group	<i>Specific group to compare with</i> scalar<character> // <i>Default:</i> NULL (optional) Both the absolute and relative folderpaths are required, as strings.
call	<i>Internal call</i> obj:<call> // <i>Default:</i> rlang::caller_env() (optional) Both the absolute and relative folderpaths are required, as strings.

**Value**

rdocx object, which can be saved with print() after loading the officer-package  
 rdocx object, which can be saved with print() after loading the officer-package

**Examples**

```
library(officer) # To save the rdocx object to disk

test_docx_b13 <-
  ex_survey |>
  embed_cat_freq_plot_docx(dep = b_1:b_3,
    showNA = "never",
    descend = TRUE,
    return_raw = FALSE,
    hide_label_if_prop_below=0,
    data_label = "count",
    data_label_decimal_symbol = ",",
    digits = 1,
    label_font_size = 12,
    main_font_size = 12,
    plot_height_multiplier = .3,
    plot_height_fixed_constant = 1,
    vertical = FALSE,
    font_family = "sans")
print(test_docx_b13, target = tempfile(fileext = ".docx"))
library(officer) # To save the rdocx object to disk
```

```

test_docx_b13 <-
  ex_survey |>
  embed_cat_freq_plot_docx(dep = b_1:b_3,
    showNA = "never",
    descend = TRUE,
    return_raw = FALSE,
    hide_label_if_prop_below=0,
    data_label = "count",
    data_label_decimal_symbol = ",",
    digits = 1,
    label_font_size = 12,
    main_font_size = 12,
    plot_height_multiplier = .3,
    plot_height_fixed_constant = 1,
    vertical = FALSE,
    font_family = "sans")
print(test_docx_b13, target = tempfile(fileext = ".docx"))

```

---

embed\_cat\_prop\_plot    *Embed Interactive Categorical Plot*

---

## Description

Embed Interactive Categorical Plot

## Usage

```

embed_cat_prop_plot(
  data,
  ...,
  dep = tidyselect::everything(),
  indep = NULL,
  colour_palette = NULL,
  mesos_group = NULL,
  html_interactive = TRUE,
  inverse = FALSE,
  call = rlang::caller_env()
)

```

## Arguments

data	<i>Survey data</i> obj:<data.frame> obj:<tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
...	<i>Dynamic dots</i> <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.

dep, indep	<i>Variable selections</i> <tidyselect> // <i>Default:</i> NULL, meaning everything for dep, nothing for indep. Columns in data. Currently allows tidyselect-syntax, which will be removed.
colour_palette	Character vector of colour codes.
mesos_group	<i>Specific group to compare with</i> scalar<character> // <i>Default:</i> NULL (optional) Both the absolute and relative folderpaths are required, as strings.
html_interactive	<i>Toggle interactive plot</i> scalar<logical> // <i>default:</i> TRUE (optional) Whether plot is to be interactive (ggiraph) or static (ggplot2).
inverse	Flag, defaults to FALSE. If TRUE, swaps x-axis and faceting.
call	<i>Internal call</i> obj:<call> // <i>Default:</i> rlang::caller_env() (optional) Both the absolute and relative folderpaths are required, as strings.

**Value**

ggplot

**Examples**

```
embed_cat_prop_plot(data = ex_survey, dep = b_1:b_3)
```

---

```
embed_cat_prop_plot_docx
```

*Create Word Report with Univariates for Categorical Columns Sharing Same Categories*

---

**Description**

Create Word Report with Univariates for Categorical Columns Sharing Same Categories

**Usage**

```
embed_cat_prop_plot_docx(
  data,
  ...,
  dep = tidyselect::everything(),
  indep = NULL,
  colour_palette = NULL,
  mesos_group = NULL,
  plot_height = 15,
  inverse = FALSE
)
```



**Arguments**

data	<i>Survey data</i> obj:<data.frame> obj:<tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
...	<i>Dynamic dots</i> <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.
dep, indep	<i>Variable selections</i> <tidyselect> // <i>Default:</i> NULL, meaning everything for dep, nothing for indep. Columns in data. Currently allows tidyselect-syntax, which will be removed.
colour_palette	Character vector of colour codes.
mesos_group	<i>Specific group to compare with</i> scalar<character> // <i>Default:</i> NULL (optional) Both the absolute and relative folderpaths are required, as strings.
plot_height	Fixed height of the plot in cm. Defaults to 15.
inverse	Flag, defaults to FALSE. If TRUE, swaps x-axis and faceting.

**Value**

rdocx object, which can be saved with print() after loading the officer-package

**Examples**

```
library(officer) # To save the rdocx object to disk

test_docx_b13 <-
  ex_survey |>
  embed_cat_prop_plot_docx(dep = b_1:b_3,
    showNA = "never",
    descend = TRUE,
    return_raw = FALSE,
    hide_label_if_prop_below=0,
    data_label = "percentage_bare",
    data_label_decimal_symbol = ", ",
    digits = 1,
    label_font_size = 12,
    main_font_size = 12,
    plot_height_multiplier = .3,
    plot_height_fixed_constant = 1,
    vertical = FALSE,
    font_family = "sans")
print(test_docx_b13, target = tempfile(fileext = ".docx"))
```

---

embed_cat_table	<i>Embed Reactable Table</i>
-----------------	------------------------------

---

## Description

Embed Reactable Table

## Usage

```
embed_cat_table(
  data,
  ...,
  dep = tidyselect::everything(),
  indep = NULL,
  mesos_group = NULL,
  call = rlang::caller_env()
)
```

## Arguments

data	<i>Survey data</i> obj:<data.frame> obj:<tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
...	<i>Dynamic dots</i> <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.
dep, indep	<i>Variable selections</i> <tidyselect> // <i>Default:</i> NULL, meaning everything for dep, nothing for indep. Columns in data. Currently allows tidyselect-syntax, which will be removed.
mesos_group	<i>Specific group to compare with</i> scalar<character> // <i>Default:</i> NULL (optional) Both the absolute and relative folderpaths are required, as strings.
call	<i>Internal call</i> obj:<call> // <i>Default:</i> rlang::caller_env() (optional) Both the absolute and relative folderpaths are required, as strings.

## Value

A reactable object. If return\_raw=FALSE, then just the table.

---

embed\_cat\_text\_html    *Creates a structured list with text interpretations for a set of variables.*

---

## Description

Creates a structured list with text interpretations for a set of variables.

## Usage

```
embed_cat_text_html(
  data,
  dep = NULL,
  indep = NULL,
  ...,
  mesos_group = NULL,
  call = rlang::caller_env()
)
```

## Arguments

data	<i>Survey data</i> obj:<data.frame> obj:<tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
dep, indep	<i>Variable selections</i> <tidyselect> // <i>Default:</i> NULL, meaning everything for dep, nothing for indep. Columns in data. Currently allows tidyselect-syntax, which will be removed.
...	<i>Dynamic dots</i> <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.
mesos_group	<i>Specific group to compare with</i> scalar<character> // <i>Default:</i> NULL (optional) Both the absolute and relative folderpaths are required, as strings.
call	<i>Internal call</i> obj:<call> // <i>Default:</i> rlang::caller_env() (optional) Both the absolute and relative folderpaths are required, as strings.

## Value

List

**Examples**

```
embed_cat_text_html(ex_survey,
  dep = tidyselect::matches("e_"),
  contents = c("intro", "mode_max", "value_max",
    "value_min", "not_used_category", "mean_max", "mean_min"),
  label_separator = " - ",
  require_common_categories = FALSE,
  n_top_bottom = 1,
  showNA = "never",
  descend = TRUE,
  return_raw = TRUE,
  hide_label_if_prop_below=0,
  data_label = "count",
  data_label_decimal_symbol = ",",
  digits = 1)
```

---

embed\_chr\_table\_html *Interactive table of text data*

---

**Description**

Interactive table of text data

**Usage**

```
embed_chr_table_html(
  data,
  dep = colnames(data),
  ...,
  mesos_group = NULL,
  call = rlang::caller_env()
)
```

**Arguments**

data	<i>Survey data</i> obj: <data.frame>   obj: <tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
dep	<i>Variable selections</i> <tidyselect> // <i>Default</i> : NULL, meaning everything for dep, nothing for by. Columns in data. Currently allows tidyselect-syntax, which will be removed.
...	<i>Dynamic dots</i> <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.

mesos_group	<i>Specific group to compare with</i> scalar<character> // <i>Default:</i> NULL (optional) Both the absolute and relative folderpaths are required, as strings.
call	<i>Internal call</i> obj:<call> // <i>Default:</i> rlang::caller_env() (optional) Both the absolute and relative folderpaths are required, as strings.

**Value**

Data frame

---

embed_sigtest	<i>Test Significance Based on Randomization Theory</i>
---------------	--

---

**Description**

Test Significance Based on Randomization Theory

**Usage**

```
embed_sigtest(data, chapter_overview, ..., call = rlang::caller_env())
```

**Arguments**

data	<i>Survey data</i> obj:<data.frame> obj:<tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
chapter_overview	<i>What goes in each chapter</i> 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.
...	<i>Dynamic dots</i> <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.
call	<i>Internal call</i> obj:<call> // <i>Default:</i> rlang::caller_env() (optional) Both the absolute and relative folderpaths are required, as strings.

**Value**

Data frame

---

 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 comma

**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	<i>File/folder name sanitizer replacing space and punctuation with underscore</i>
--------------------	---

---

**Description**

File/folder name sanitizer replacing space and punctuation with underscore

**Usage**

```
filename_sanitizer(x, max_chars = NA_integer_)
```

**Arguments**

x	Character vector of file/folder names
max_chars	Maximum character length

**Value**

Character vector of same length as x

**Examples**

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

---

get_colour_palette	<i>Provide A Colour Set for A Number of Requested Colours</i>
--------------------	---

---

**Description**

Possibly using colour\_palette\_nominal if available. If not sufficient, uses a set palette from RColorBrewer.

**Usage**

```
get_colour_palette(
  data,
  col_pos,
  colour_palette_nominal = NULL,
  colour_palette_ordinal = NULL,
  colour_na = NULL,
  categories_treated_as_na = NULL,
  call = rlang::caller_env()
)
```



**Arguments**

data	<i>Survey data</i> obj:<data.frame> obj:<tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
col_pos	Character vector of column names for which colours will be found.
colour_palette_nominal, colour_palette_ordinal	<i>User specified colour set</i> vector<character> // default: NULL (optional) User-supplied default palette, excluding colour_na.
colour_na	<i>Colour for NA category</i> scalar<character> // default: NULL (optional) Colour as a single string for NA values.
categories_treated_as_na	<i>NA categories</i> vector<character> // default: NULL (optional) Categories that should be treated as NA.
call	<i>Internal call</i> obj:<call> // Default: rlang::caller_env() (optional) Both the absolute and relative folderpaths are required, as strings.

**Value**

A colour set as character vector, where NA has the colour\_na, and the rest are taken from colour\_palette\_nominal if available.

**Examples**

```
get_colour_palette(ex_survey, col_pos=c("b_1", "b_2"))
get_colour_palette(ex_survey, col_pos=c("b_1", "b_2"),
  colour_palette_nominal = c("red", "blue", "orange"))
```

---

get\_colour\_set

*Provide A Colour Set for A Number of Requested Colours*


---

**Description**

Possibly using colour\_palette\_nominal if available. If not sufficient, uses a set palette from RColorBrewer.

**Usage**

```
get_colour_set(
  x,
  common_data_type = "factor",
  colour_palette_nominal = NULL,
  colour_palette_ordinal = NULL,
  colour_na = NULL,
  colour_2nd_binary_cat = NULL,
  ordinal = FALSE,
  categories_treated_as_na = NULL,
  call = rlang::caller_env()
)
```

**Arguments**

**x** Vector for which colours will be found.

**common\_data\_type**  
*factor or ordered data type*  
 scalar<character> // *default: factor* (optional)  
 Currently only supports factor and ordered.

**colour\_palette\_nominal, colour\_palette\_ordinal**  
*User specified colour set*  
 vector<character> // *default: NULL* (optional)  
 User-supplied default palette, excluding colour\_na.

**colour\_na** *Colour for NA category*  
 scalar<character> // *default: NULL* (optional)  
 Colour as a single string for NA values.

**colour\_2nd\_binary\_cat**  
*Colour for second binary category*  
 scalar<character> // *default: "#ffffff"* (optional)  
 Colour for second category in binary variables. Often useful to hide this.

**ordinal** scalar<logical> // *default: FALSE* (optional)  
 Is palette ordinal?

**categories\_treated\_as\_na**  
*NA categories*  
 vector<character> // *default: NULL* (optional)  
 Categories that should be treated as NA.

**call** *Internal call*  
 obj:<call> // *Default: rlang::caller\_env()* (optional)  
 Both the absolute and relative folderpaths are required, as strings.

**Value**

A colour set as character vector, where NA has the colour\_na, and the rest are taken from colour\_palette\_nominal if available.

**Examples**

```
get_colour_set(x=1:4)
```

---

```
get_data_label_opts
```

*Get Valid Data Labels for Figures and Tables*

---

**Description**

Get Valid Data Labels for Figures and Tables

**Usage**

```
get_data_label_opts()
```

**Value**

Character vector

---

```
handpick
```

*Return character vector of manually picked data columns.*

---

**Description**

Only works in an interactive session. Copies to the clipboard if `to_clipboard = TRUE`, and if clipboard is available on system.

**Usage**

```
handpick(data, to_clipboard = FALSE, bare = FALSE)
```

**Arguments**

<code>data</code>	A dataset; data frame.
<code>to_clipboard</code>	Boolean. Defaults to <code>FALSE</code> . If <code>TRUE</code> , overwrites what you already have copied. Use with caution.
<code>bare</code>	Flag, defaults to <code>FALSE</code> . If <code>TRUE</code> , returns a tidyselect-style bare list of columns, without quotation marks.

**Value**

Character vector. Prints to console.

**Examples**

```
if(interactive()) handpick(mtcars)
```

---

hex_bw	<i>Identify Suitable Font Given Background Hex Colour</i>
--------	---

---

**Description**

Code is taken from XXX.

**Usage**

```
hex_bw(hex_code)
```

**Arguments**

hex\_code          Colour in hex-format.

**Value**

Colours in hex-format, either black or white.

**Examples**

```
hex_bw("#0dadfd")
```

---

is_colour	<i>Are All Colours in Vector Valid Colours</i>
-----------	--

---

**Description**

As title says. From: (<https://stackoverflow.com/a/13290832/3315962>)

**Usage**

```
is_colour(x)
```

**Arguments**

x                  Character vector of colours in hex-format.

**Value**

Logical, or error.

**Examples**

```
is_colour(c("#ff00ff", "#010101"))
```

---

 list\_available\_element\_types

*List All Valid Names of The Elements Argument*


---

**Description**

List All Valid Names of The Elements Argument

**Usage**

```
list_available_element_types(valid_only = TRUE)
```

**Arguments**

`valid_only` Only return implemented elements, or all (planned) elements. Note that \*\_docx- elements have limited storage support and are hence not included in examples in this package as minor (unimportant) warnings will occur.

**Value**

Character vector of valid names.

---

 omitted\_recoder\_df *Recode Missing By Type of Missingness*


---

**Description**

Useful for item difficulty estimation according to Mislevy's recommendation. Also allowing for escaping rows with all missingness (typically not administered).

**Usage**

```
omitted_recoder_df(
  df,
  accept_vector = FALSE,
  skipped = 0L,
  not_administered = NA_integer_,
  all_missing = NA_integer_
)
```

**Arguments**

df	Data frame, or vector. Must be a dataframe, not a matrix, in this function. Only include item variables.
accept_vector	Handles vectors if accept_vector=TRUE. Set to false to avoid accidents when using function per block and there is just one item in the block.
skipped	What to replace skipped values with
not_administered	What to replace not administered values with.
all_missing	What to replace values in rows with all missing with.

**Value**

A data.frame (or vector, if input is vector and accept\_vector=TRUE) with recoded cells.

**Examples**

```
# Original data
input <- stats::setNames(as.data.frame(matrix(c(
  1,0,1,0,1, # All present
  NA,0,1,0,1, # First missing
  NA,NA,1,0,1, # First two missing
  1,0,NA,0,1, # One in middle missing
  1,NA,NA,NA,1, # All in the middle missing
  1,0,1,0,NA, # Last one missing
  1,0,1,NA,NA, # Last two missing
  1,0,NA,NA,NA, # Last three missing
  NA,NA,NA,NA,NA # All missing
), nrow = 9, byrow = TRUE)), nm=stringi::stri_c(ignore_null=TRUE, "X", 1:5))
# What should be the output for item estimation according to Mislevy
# Skipped=> 0, not_administered=>NA, all_missing=>NA
y_i <- stats::setNames(as.data.frame(matrix(c(
  1,0,1,0,1, # All present
  0,0,1,0,1, # First missing
  0,0,1,0,1, # First two missing
  1,0,0,0,1, # One in middle missing
  1,0,0,0,1, # All in the middle missing
  1,0,1,0,0, # Last one missing
  1,0,1,0,NA, # Last two missing
  1,0,0,NA,NA, # Last three missing
  NA,NA,NA,NA,NA # All missing
), nrow = 9, byrow = TRUE)), nm=stringi::stri_c(ignore_null=TRUE, "X", 1:5))

# What should be the output for person estimation according to Mislevy
# Skipped=> 0, not_administered=>NA, all_missing=>NA
y_p <- stats::setNames(as.data.frame(matrix(c(
  1,0,1,0,1, # All present
  0,0,1,0,1, # First missing
  0,0,1,0,1, # First two missing
  1,0,0,0,1, # One in middle missing
  1,0,0,0,1, # All in the middle missing
```

```

1,0,1,0,0, # Last one missing
1,0,1,0,0, # Last two missing
1,0,0,0,0, # Last three missing
0,0,0,0,0 # All missing
), nrow = 9, byrow = TRUE)), nm=stringi::stri_c(ignore_null=TRUE, "X", 1:5))
# Recoding for counting skipped, not_administered, all_missing, etc
# Skipped=> 99, not_administered=>999, all_missing=>9999
y_info <- stats::setNames(as.data.frame(matrix(c(
1,0,1,0,1, # All present
99,0,1,0,1, # First missing
99,99,1,0,1, # First two missing
1,0,99,0,1, # One in middle missing
1,99,99,99,1, # All in the middle missing
1,0,1,0,99, # Last one missing
1,0,1,99,999, # Last two missing
1,0,99,999,999, # Last three missing
9999,9999,9999,9999,9999 # All missing
), nrow = 9, byrow = TRUE)), nm=stringi::stri_c(ignore_null=TRUE, "X", 1:5))

y_i2 <- omitted_recoder_df(input) #Mislevy item estimation
y_p2 <- omitted_recoder_df(input, skipped = 0L, #Mislevy person estimation
                           not_administered = 0L, all_missing = 0L)
y_info2 <- omitted_recoder_df(input, skipped = 99,
                              not_administered = 999, all_missing = 9999)

identical(y_i, y_i2)
identical(y_p, y_p2)
identical(y_info, y_info2)
## Not run:
omitted_recoder_df(input[,4]) # Should fail

## End(Not run)
identical(omitted_recoder_df(input[,4], accept_vector=TRUE),
          c(0,0,0,0,0,0,0,NA,NA))
identical(omitted_recoder_df(input[,4, drop=FALSE]),
          input[,4, drop=FALSE]) # Output should equal input

```

---

```
post_render_docx_img_replacer
```

*Post-Render All docx-files Found in The Output Folder To Replace Images with Mscharts*

---

## Description

Post-Render All docx-files Found in The Output Folder To Replace Images with Mscharts

## Usage

```

post_render_docx_img_replacer(
  site_dir = fs::path(Sys.getenv("QUARTO_PROJECT_OUTPUT_DIR"), "reports", "report"),
  site_mesos_dir = fs::path(site_dir, "mesos"),

```

```

chart_dir = fs::path(Sys.getenv("QUARTO_PROJECT_OUTPUT_DIR"), "..", "reports",
  "report"),
chart_mesos_dir = fs::path(chart_dir, "mesos"),
delete_mschart_files = FALSE
)

```

### Arguments

site\_dir, site\_mesos\_dir, chart\_dir, chart\_mesos\_dir  
 String. Paths to the site (and mesos subfolder) with the docx files where the images are to be replaced, and paths to where to find the docx files with mscharts to be replaced with can be found.

delete\_mschart\_files  
 Flag. Whether to delete the mschart docx files after successful copying into the docx files. Defaults to FALSE.

### Value

Nothing, side-effects only.

---

prepare_chunk	<i>Internal function to prepare a chunk for a Quarto report.</i>
---------------	--

---

### Description

Internal function to prepare a chunk for a Quarto report.

### Usage

```
prepare_chunk(element_name, ...)
```

### Arguments

element\_name    The name of the element to be prepared.  
 ...             Dynamic dots passed onto the methods.

### Value

Returns a text string for the chunk, but also has side-effects for generating files.



---

```
prepare_safe_render
```

*Convenience function to prepare, copy and render website*

---

### Description

Rendering a website on a Sharepoint/OneDrive location can be a pain due to long filepaths above 260 characters, which causes read errors. This function simplifies things by copying a site from `local_basepath` to a `site_basepath`, sets up basic access authentication files (either Netlify's `_headers` file or regular Apache `.htaccess/.htpasswd` files), as well as optionally copying in an existing Netlify `_publish.yaml`-file.

### Usage

```
prepare_safe_render(
  remote_basepath = "/home/",
  from_folders = NULL,
  from_files = NULL,
  local_basepath,
  site,
  rel_path_base_to_parent_of_user_restricted_folder = "Reports",
  overwrite = FALSE,
  prompt = TRUE
)
```

### Arguments

<code>remote_basepath</code>	String. FTP servers often have a subdomain path location different from <code>"/home/"</code> . Adjust this here.
<code>from_folders, from_files</code>	Character vector of folders and files to copy into the site path.
<code>local_basepath</code>	String. Path to where your QMD-files are located (the site to be rendered).
<code>site</code>	String. Path to where to copy folders and files to setup what Quarto needs to build a project.
<code>rel_path_base_to_parent_of_user_restricted_folder</code>	String. Path going from basepath to the folder containing folders to password-protect.
<code>overwrite</code>	Flag. Defaults to <code>FALSE</code> to ensure you know what you are doing. If <code>TRUE</code> , will delete all files and folders in site!
<code>prompt</code>	Flag. Whether to ask the user if they are certain. Defaults to <code>TRUE</code> .

### Value

`local_basepath`

---

quarto\_render\_saros     *Simple wrapper for quarto::quarto\_render() that temporarily sets LC\_ALL="C" and takes the processing time*

---

### Description

Simple wrapper for quarto::quarto\_render() that temporarily sets LC\_ALL="C" and takes the processing time

### Usage

```
quarto_render_saros(site_path, ...)
```

### Arguments

site_path	String, path to render.
...	Dynamic dots forwarded to quarto::quarto_render

### Value

Nothing

---

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. <i>default:</i> settings.yaml
------	--

### Value

The defaults as a yaml-object.

### Examples

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

---

 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,
  ...,
  progress = TRUE,
  variable_group_dep = ".variable_group_dep",
  call = rlang::caller_env()
)
```

## Arguments

chapter\_overview

*What goes in each 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> // Required

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

...

*Dynamic dots*

<dynamic-dots>

Arguments forwarded to the corresponding functions that create the elements.

progress

*Whether to display progress message*

scalar<logical> // default: TRUE

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.

call

*Internal call*

obj:<call> // Default: rlang::caller\_env() (optional)

Both the absolute and relative folderpaths are required, as strings.

**Value**

Grouped tibble.

**Examples**

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

ref_df2 <- refine_chapter_overview(chapter_overview = ex_survey_ch_overview,
                                   data = ex_survey, hide_bi_entry_if_sig_above=.05)
```

---

remove\_label\_parts      *Remove regex pattern from variable labels*

---

**Description**

Remove regex pattern from variable labels

**Usage**

```
remove_label_parts(  
  data,  
  pattern = NULL,  
  replacement = "",  
  data_type_criterion = NULL  
)
```

**Arguments**

data	Data.frame
pattern	String, regex pattern.
replacement	String, replacement for pattern.
data_type_criterion	Can be a function that specifies what kind of variables to select, e.g. is.factor

**Value**

Data

**Examples**

```
remove_label_parts(ex_survey, pattern = " living in")
```

---

`remove_special_chars_in_labels`*Remove Special Characters (<,>) in Variable Labels*

---

**Description**

Remove Special Characters (<,>) in Variable Labels

**Usage**

```
remove_special_chars_in_labels(df)
```

**Arguments**

`df`                      Data frame

**Value**

A data frame

---

`rename_by_labels`*Rename Dataset Columns by Labels.*

---

**Description**

Occasionally dataframe columns have not been named logically and consistent in the software where the data originates. This function renames variable names based on patterns in the variable labels, after ignoring some stop words.

Occasionally dataframe columns have not been named logically and consistent in the software where the data originates. This function renames variable names based on patterns in the variable labels, after ignoring some stop words.

**Usage**

```
rename_by_labels(  
  data,  
  label_sep = " - ",  
  sort_var = c("pos", "variable", "label"),  
  new_var_sep = "_",  
  stop_words = NULL  
)
```

```
rename_by_labels(  
  data,  
  label_sep = " - ",
```

```

    sort_var = c("pos", "variable", "label"),
    new_var_sep = "_",
    stop_words = NULL
  )

```

### Arguments

data	Dataset.
label_sep	The separator between group part and unique part of label.
sort_var	When numbering variables within a group, what to sort by? pos is original position in dataset,
new_var_sep	When creating new variables, how to glue together variable group name prefix and numbering?
stop_words	Words to ignore in label when abbreviating label to name.

### Value

Data with renamed variable names.

Data with renamed variable names.

### Examples

```

rename_by_labels(ex_survey)
rename_by_labels(ex_survey)

```

---

render\_full\_reports    *Post-render rendering of complete reports*

---

### Description

If one wishes to render complete reports that are linked to in a website, but not listed among the chapters on the sidebar menu, one can make these with filenames starting with underscores (\_) which will make them not listed in said menu. However, this will also mean they are not rendered in a Quarto (Website) project. This function, which can be called within a post-script (see example below), will render only these after the ordinary rendering of the project, and copied to the `_site` folder.

### Usage

```

render_full_reports(
  files = NULL,
  path,
  processable_path = file.path(path, "Reports"),
  site_path = file.path(path, "_site"),
  resource_paths = file.path(path, c("_extensions", "_images")),
  warn_on_file_error = FALSE,
  ...
)

```

**Arguments**

files	Optional character vector of report files (qmd). Can be obtained within a project by <code>Sys.getenv("QUARTO_PROJECT_OUTPUT_FILES")</code>
path	If no files are given, a path to the root folder of the local "site".
processable_path	Path to where report files can be (recursively) found.
site_path	Path to <code>_site</code>
resource_paths	Paths to where <code>_extensions</code> and <code>_images</code> folders can be found and copied to wherever needed
warn_on_file_error	If TRUE, will collect warnings if a file fails to render or be copied. If FALSE (default), will stop the rendering process.
...	Additional arguments passed to <code>quarto::render()</code>

**Value**

Returns invisibly a character vector of processed files.

---

```
replace_docx_imgs_with_mscharts
```

*Replace Images in A Quarto-Generated docx-file with Mschart docx-files*

---

**Description**

Replace Images in A Quarto-Generated docx-file with Mschart docx-files

**Usage**

```
replace_docx_imgs_with_mscharts(
  main_file,
  chart_dir = fs::path_ext_remove(main_file),
  delete_mschart_files = FALSE
)
```

**Arguments**

main_file	docx file where images are to be replaced with mscharts
chart_dir	Directory path to where the mscharts as docx files can be found.
delete_mschart_files	Flag. Whether to delete the mschart docx files after insertion. Defaults to FALSE.

**Value**

String, path to input file.

---

replace\_stata\_labels    *Replace Stata Labels*

---

**Description**

If no replacement exists, keeps the original.

**Usage**

```
replace_stata_labels(  
  data,  
  df_new_labels,  
  var_name_col = "name",  
  var_label_col = "vallab_full",  
  trim = c("both", "all"),  
  overwrite = TRUE  
)
```

**Arguments**

`data`                    data frame for which labels will be replaced

`df_new_labels`        a data frame with a `var_name_col` and a `var_label_col`

`var_name_col`, `var_label_col`  
                         variable names (as strings) for the lookup of variable names in data, and the  
                         corresponding variable label.

`trim`                    string, trim "both" sides (default), "left" or "right".

`overwrite`              Flag. Whether to remove label if it already exists.

**Value**

data, with variable labels replaced

---

sanitize\_labels            *Sanitize labels originating from e.g. Qualtrics for use in Saros*

---

**Description**

This function is quite specific to a few problems, users might find it lacking in functionality.



**Usage**

```

sanitize_labels(
  data,
  sep = " - ",
  multi_sep_replacement = ": ",
  replace_ascii_with_utf = FALSE,
  questions = NULL
)

```

**Arguments**

data	data.frame or tibble
sep	String, separates main question from subquestion
multi_sep_replacement	String. If multiple sep are found, replace the first ones with this.
replace_ascii_with_utf	Flag. If TRUE, downloads a list from W3 used to convert html characters as ASCII to UTF8.
questions	Data frame with questions obtained from <code>qualtRics::survey_questions()</code>

**Value**

Identical data.frame as input, with only variable labels changed.

**Examples**

```
sanitize_labels(ex_survey)
```

---

```
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")
)

```

### 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.

### Value

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

---

subset_vector	<i>Given Ordered Integer Vector, Return Requested Set.</i>
---------------	--

---

**Description**

Useful for identifying which categories are to be collected.

**Usage**

```
subset_vector(
  vec,
  set = c(".top", ".upper", ".mid_upper", ".lower", ".mid_lower", ".bottom", ".spread"),
  spread_n = NULL,
  sort = FALSE
)
```

**Arguments**

vec	A vector of any type.
set	A character string, one of c(".top", ".upper", ".mid_upper", ".lower", ".mid_lower", ".bottom")
spread_n	The number of values to extract when set is "spread".
sort	Whether to sort the output, defaults to FALSE.

**Value**

Selected set of vector.

**Examples**

```
subset_vector(vec=1:7, set=".mid_lower")
```

---

summarize_data	<i>Summarize a survey dataset for use in tables and graphs</i>
----------------	--

---

**Description**

Summarize a survey dataset for use in tables and graphs

**Usage**

```
summarize_data(
  data,
  ...,
  dep = colnames(data),
  indep = NULL,
  call = rlang::caller_env()
)
```

**Arguments**

data	<i>Survey data</i> obj:<data.frame> obj:<tbl_df> // Required A data frame (or a srvyr-object) with the columns specified in the chapter_overview 'dep_cat', etc columns.
...	<i>Dynamic dots</i> <dynamic-dots> Arguments forwarded to the corresponding functions that create the elements.
dep, indep	<i>Variable selections</i> <tidyselect> // <i>Default:</i> NULL, meaning everything for dep, nothing for indep. Columns in data. Currently allows tidyselect-syntax, which will be removed.
call	<i>Internal call</i> obj:<call> // <i>Default:</i> rlang::caller_env() (optional) Both the absolute and relative folderpaths are required, as strings.

**Value**

Dataset

---

 swap\_label\_colnames    *Swap Dataset Columns and Labels*


---

**Description**

Columns not containing labels will remain unaffected, and warning given.

Columns not containing labels will remain unaffected, and warning given.

**Usage**

```
swap_label_colnames(data)
```

```
swap_label_colnames(data)
```

**Arguments**

data	Data frame
------	------------

**Value**

Data.frame

Data.frame

**Examples**

```
swap_label_colnames(mtcars)
```

```
swap_label_colnames(mtcars)
```

---

`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)
```

**Arguments**

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

**Value**

The defaults as a yaml-object.

**Examples**

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

# Index

- \* **datasets**
  - ex\_survey, [30](#)
  - ex\_survey\_ch\_overview, [31](#)
- attach\_qualtrics\_labels, [3](#)
- center\_string, [4](#)
- col\_to\_binaries, [4](#)
- combn\_upto, [5](#)
- create\_caption, [6](#)
- create\_email\_credentials, [7](#)
- crosstable3, [8](#)
- draft\_report, [8](#)
- embed\_cat\_freq\_plot, [20](#)
- embed\_cat\_freq\_plot\_docx, [21](#)
- embed\_cat\_prop\_plot, [23](#)
- embed\_cat\_prop\_plot\_docx, [24](#)
- embed\_cat\_table, [26](#)
- embed\_cat\_text\_html, [27](#)
- embed\_chr\_table\_html, [28](#)
- embed\_sigtest, [29](#)
- ex\_survey, [30](#)
- ex\_survey\_ch\_overview, [31](#)
- filename\_sanitizer, [32](#)
- get\_colour\_palette, [32](#)
- get\_colour\_set, [33](#)
- get\_data\_label\_opts, [35](#)
- handpick, [35](#)
- hex\_bw, [36](#)
- is\_colour, [36](#)
- list\_available\_element\_types, [37](#)
- omitted\_recoder\_df, [37](#)
- post\_render\_docx\_img\_replacer, [39](#)
- prepare\_chunk, [40](#)
- prepare\_safe\_render, [41](#)
- quarto\_render\_saros, [42](#)
- read\_default\_draft\_report\_args, [42](#)
- refine\_chapter\_overview, [43](#)
- remove\_label\_parts, [44](#)
- remove\_special\_chars\_in\_labels, [45](#)
- rename\_by\_labels, [45](#)
- render\_full\_reports, [46](#)
- replace\_docx\_imgs\_with\_mscharts, [47](#)
- replace\_stata\_labels, [48](#)
- sanitize\_labels, [48](#)
- setup\_access\_restrictions, [49](#)
- subset\_vector, [51](#)
- summarize\_data, [51](#)
- swap\_label\_colnames, [52](#)
- write\_default\_draft\_report\_args, [53](#)