

# Package ‘volker’

February 22, 2024

**Type** Package

**Title** High-Level Functions for Tabulating, Charting and Reporting  
Survey Data

**Version** 1.0.2

**Date** 2024-02-03

**Description** Craft polished tables and plots in Markdown reports.  
Simply choose whether to treat your data as counts or metrics,  
and the package will automatically generate well-designed default tables and plots for you.  
Boiled down to the basics, with labeling features and simple interactive reports.  
All functions are 'tidyverse' compatible.

**URL** <https://github.com/strohne/volker>

**BugReports** <https://github.com/strohne/volker/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**LazyData** true

**Imports** stats, rlang, lifecycle, glue, tibble, dplyr, tidyr,  
tidyselect, ggplot2 (>= 2.2.1), scales, base64enc, stringr,  
purrr, magrittr, forcats, skimr, knitr, kableExtra, rmarkdown,  
janitor, psych

**Depends** R (>= 4.2)

**Suggests** tidyverse, remotes, usethis, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Jakob Jünger [aut, cre, cph] (<<https://orcid.org/0000-0003-1860-6695>>),  
Henrieke Kotthoff [ctb],  
Chantal Gärtner [ctb] (<<https://orcid.org/0000-0002-3653-6013>>)

**Maintainer** Jakob Jünger <jakob.juenger@uni-muenster.de>

**Repository** CRAN

**Date/Publication** 2024-02-22 21:00:02 UTC

## R topics documented:

chatgpt . . . . .	2
codebook . . . . .	3
data_clean . . . . .	3
html_report . . . . .	4
idx_add . . . . .	5
labs_apply . . . . .	5
labs_clear . . . . .	6
labs_restore . . . . .	7
labs_store . . . . .	8
plot_counts . . . . .	8
plot_metrics . . . . .	9
report_counts . . . . .	10
report_metrics . . . . .	11
tab_counts . . . . .	12
tab_metrics . . . . .	13
<b>Index</b>	<b>15</b>

---

chatgpt

*ChatGPT Adoption Dataset CG-GE-APR22*

---

### Description

A small random subset of data from a survey about ChatGPT adoption. The survey was conducted in April 2023 within the population of Germany Internet users.

### Usage

chatgpt

### Format

chatgpt:

A data frame with 101 rows and 19 columns:

**case** A running case number

**adopter** Adoption groups inspired by Roger's innovator typology.

**use\_** Columns starting with use contain data about ChatGPT usage in different contexts.

**cg\_activities** Text answers to the question, what the respondents do with ChatGPT.

**cg\_adoption\_** A scale consisting of items about advantages, fears, and social aspects. The scales match theoretical constructs inspired by Roger's diffusion model and Davis' Technology Acceptance Model

**sd\_** Columns starting with sd contain sociodemographics of the respondents.

### Details

Call `codebook(volker::chatgpt)` to see the items and answer options.

**Source**

Communication Department of the University of Münster ([gehrau@uni-muenster.de](mailto:gehrau@uni-muenster.de)).

---

codebook	<i>Get variable labels from their comment attributes</i>
----------	--

---

**Description**

Get variable labels from their comment attributes

**Usage**

```
codebook(data, cols)
```

**Arguments**

data	A tibble
cols	A tidy variable selections to filter specific columns

**Value**

A tibble with the columns: - `item_name`: The column name. - `item_group`: First part of the column name, up to an underscore. - `item_class`: The last class value of an item (e.g. numeric, factor). - `item_label`: The comment attribute of the column. - `value_name`: In case a column has numeric attributes, the attribute names - `value_label`: In case a column has numeric attributes or T/F-attributes, the attribute values. In case a column has a levels attribute, the levels.

**Examples**

```
volker::codebook(volker::chatgpt)
```

---

data_clean	<i>Prepare dataframe for tabs, plots, and index operations</i>
------------	--

---

**Description**

The tibble remembers whether it was already cleaned and the cleaning plan is only performed once in the first call.

**Usage**

```
data_clean(data, plan = "sosci", ...)
```

**Arguments**

<code>data</code>	Data frame
<code>plan</code>	The cleaning plan. By now, only "sosci" is supported. See <a href="#">data_clean_sosci</a> .
<code>...</code>	Other parameters passed to the appropriate cleaning function

**Value**

Cleaned data frame with `vlkr_df` class

**Examples**

```
ds <- volker::chatgpt
ds <- data_clean(ds)
```

---

html\_report

*Volker style HTML document format*

---

**Description**

Based on the standard theme, tweaks the pill navigation to switch between tables and plots. To use the format, in the header of your Markdown document, set `output: volker::html_report`.

**Usage**

```
html_report(...)
```

**Arguments**

`...` Additional arguments passed to `html_document`

**Value**

R Markdown output format

**Examples**

```
## Not run:
# Add `volker::html_report` to the output options of your Markdown document:
#
# ```
# ---
# title: "How to create reports?"
# output: volker::html_report
# ---
# ```

## End(Not run)
```

---

idx_add	<i>Calculate the mean value of multiple items</i>
---------	---

---

**Description**

Calculate the mean value of multiple items

**Usage**

```
idx_add(data, cols, newcol = NULL, negative = FALSE, clean = TRUE)
```

**Arguments**

data	A dataframe
cols	A tidy selection of item columns
newcol	Name of the index as a character value. Set to NULL (default) to automatically build a name from the common column prefix, prefixed with "idx_"
negative	If FALSE (default), negative values are recoded as missing values.
clean	Prepare data by <a href="#">data_clean</a> .

**Value**

The input tibble with an additional column that contains the index values. The column contains the result of the alpha calculation in the attribute named "psych.alpha".

**Examples**

```
ds <- volker::chatgpt
volker::idx_add(ds, starts_with("cg_adoption"))
```

---

labs_apply	<i>Set variable labels by setting their comment attributes</i>
------------	--

---

**Description**

Set variable labels by setting their comment attributes

**Usage**

```
labs_apply(data, codes, cols = NULL, values = TRUE)
```

**Arguments**

data	A tibble
codes	A tibble in <code>codebook</code> format. To set column labels, use <code>item_name</code> and <code>item_label</code> columns.
cols	A tidy column selection. Set to <code>NULL</code> (default) to apply to all columns found in the codebook. Restricting the columns is helpful when you want to set value labels. In this case, provide a tibble with <code>value_name</code> and <code>value_label</code> columns and specify the columns that should be modified.
values	If <code>TRUE</code> (default), sets value labels. - For factors: Factor levels and order are retrieved from the <code>value_label</code> column. - For item values: they are retrieved from both the columns <code>value_name</code> and <code>value_label</code> in your codebook.

**Value**

A tibble with new labels

**Examples**

```
library(tibble)
library(volker)

newlabels <- tribble(
  ~item_name,           ~item_label,
  "cg_adoption_advantage_01", "Allgemeine Vorteile",
  "cg_adoption_advantage_02", "Finanzielle Vorteile",
  "cg_adoption_advantage_03", "Vorteile bei der Arbeit",
  "cg_adoption_advantage_04", "Macht mehr Spaß"
)

volker::chatgpt %>%
  labs_apply(newlabels) %>%
  tab_metrics(starts_with("cg_adoption_advantage_"))
```

---

labs\_clear

*Remove all comments from the selected columns*


---

**Description**

Remove all comments from the selected columns

**Usage**

```
labs_clear(data, cols, labels = NULL)
```

**Arguments**

data	A tibble
cols	Tidyselct columns
labels	The attributes to remove. NULL to remove all attributes except levels and class

**Value**

A tibble with comments removed

**Examples**

```
library(volker)
volker::chatgpt |>
  labs_clear()
```

---

labs_restore	<i>Restore labels from the codebook store in the codebook attribute.</i>
--------------	--

---

**Description**

You can store labels before mutate operations by calling [labs\\_store](#).

**Usage**

```
labs_restore(data, cols = NULL, values = TRUE)
```

**Arguments**

data	A data frame
cols	A tidyselct column selection
values	If TRUE (default), restores value labels in addition to item labels. Item labels correspond to columns, value labels to values in the columns.

**Value**

A data frame

**Examples**

```
library(dplyr)
library(volker)

volker::chatgpt |>
  labs_store() |>
  mutate(sd_age = 2024 - sd_age) |>
  labs_restore() |>
  tab_metrics(sd_age)
```

---

labs_store	<i>Get the current codebook and store it in the codebook attribute.</i>
------------	---

---

### Description

You can restore the labels after mutate operations by calling [labs\\_restore](#).

### Usage

```
labs_store(data)
```

### Arguments

data            A data frame

### Value

A data frame

### Examples

```
library(dplyr)
library(volker)

volker::chatgpt |>
  labs_store() |>
  mutate(sd_age = 2024 - sd_age) |>
  labs_restore() |>
  tab_metrics(sd_age)
```

---

plot_counts	<i>Output a frequency plot</i>
-------------	--------------------------------

---

### Description

The type of frequency plot depends on the number of selected columns:

- One column: see [plot\\_counts\\_one](#)
- Multiple columns: see [plot\\_counts\\_items](#)
- One column and one grouping column: see [plot\\_counts\\_one\\_grouped](#)
- Multiple columns and one grouping column: see [plot\\_counts\\_items\\_grouped](#)

### Usage

```
plot_counts(data, cols, col_group = NULL, clean = TRUE, ...)
```

**Arguments**

data	A data frame
cols	A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as <code>starts_with()</code>
col_group	Optional, a grouping column. The column name without quotes.
clean	Prepare data by <a href="#">data_clean</a> .
...	Other parameters passed to the appropriate plot function

**Value**

A ggplot2 plot object

**Examples**

```
library(volker)
data <- volker::chatgpt

plot_counts(data, sd_gender)
```

---

plot\_metrics

*Output a plot with distribution parameters such as the mean values*


---

**Description**

The table type depends on the number of selected columns:

- One column: see [plot\\_metrics\\_one](#)
- Multiple columns: see [plot\\_metrics\\_items](#)
- One column and one grouping column: see [plot\\_metrics\\_one\\_grouped](#)
- Multiple columns and one grouping column: see [plot\\_metrics\\_items\\_grouped](#)

**Usage**

```
plot_metrics(data, cols, col_group = NULL, clean = TRUE, ...)
```

**Arguments**

data	A data frame
cols	A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as <code>starts_with()</code> .
col_group	Optional, a grouping column (without quotes).
clean	Prepare data by <a href="#">data_clean</a> .
...	Other parameters passed to the appropriate plot function

**Value**

A ggplot object

**Examples**

```
library(volker)
data <- volker::chatgpt

plot_metrics(data, sd_age)
```

---

 report\_counts

*Create table and plot for categorical variables*


---

**Description**

Depending on your column selection, different types of plots and tables are generated. See [plot\\_counts](#) and [tab\\_counts](#).

**Usage**

```
report_counts(
  data,
  cols,
  col_group = NULL,
  index = TRUE,
  numbers = NULL,
  title = TRUE,
  close = TRUE,
  clean = TRUE,
  ...
)
```

**Arguments**

data	A data frame
cols	A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as <code>starts_with()</code> .
col_group	Optional, a grouping column (without quotes).
index	When the cols contain items on a metric scale (as determined by <a href="#">get_direction</a> ), an index will be calculated using the 'psych' package. Set to FALSE to suppress index generation.
numbers	The numbers to print on the bars: "n" (frequency), "p" (percentage) or both. Set to NULL to remove numbers.
title	A character providing the heading or TRUE (default) to output a heading. Classes for tabset pills will be added.

close	Whether to close the last tab (default value TRUE) or to keep it open. Keep it open to add further custom tabs by adding headers on the fifth level in Markdown (e.g. ##### Method)
clean	Prepare data by <a href="#">data_clean</a> .
...	Parameters passed to the plot and tab functions.

### Details

For item batteries, an index is calculated and reported. When used in combination with the Markdown-template "html\_report", the different parts of the report are grouped under a tabsheet selector.

### Value

A volker report object

### Examples

```
library(volker)
data <- volker::chatgpt

report_counts(data, sd_gender)
```

---

report_metrics	<i>Create table and plot for metric variables</i>
----------------	---

---

### Description

Depending on your column selection, different types of plots and tables are generated. See [plot\\_metrics](#) and [tab\\_metrics](#).

### Usage

```
report_metrics(
  data,
  cols,
  col_group = NULL,
  ...,
  index = TRUE,
  title = TRUE,
  close = TRUE,
  clean = TRUE
)
```

**Arguments**

data	A data frame
cols	A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as <code>starts_with()</code> .
col_group	Optional, a grouping column (without quotes).
...	Parameters passed to the plot and tab functions.
index	When the cols contain items on a metric scale (as determined by <a href="#">get_direction</a> ), an index will be calculated using the 'psych' package. Set to FALSE to suppress index generation.
title	A character providing the heading or TRUE (default) to output a heading. Classes for tabset pills will be added.
close	Whether to close the last tab (default value TRUE) or to keep it open. Keep it open to add further custom tabs by adding headers on the fifth level in Markdown (e.g. ##### Method)
clean	Prepare data by <a href="#">data_clean</a> .

**Details**

For item batteries, an index is calculated and reported. When used in combination with the Markdown-template "html\_report", the different parts of the report are grouped under a tabsheet selector.

**Value**

A volker report object

**Examples**

```
library(volker)
data <- volker::chatgpt

report_metrics(data, sd_age)
```

---

tab_counts	<i>Output a frequency table</i>
------------	---------------------------------

---

**Description**

The type of frequency table depends on the number of selected columns:

- One column: see [tab\\_counts\\_one](#)
- Multiple columns: see [tab\\_counts\\_items](#)
- One column and one grouping column: see [tab\\_counts\\_one\\_grouped](#)
- Multiple columns and one grouping column: see [tab\\_counts\\_items\\_grouped](#)

**Usage**

```
tab_counts(data, cols, col_group = NULL, clean = TRUE, ...)
```

**Arguments**

data	A data frame
cols	A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as <code>starts_with()</code>
col_group	Optional, a grouping column. The column name without quotes.
clean	Prepare data by <a href="#">data_clean</a> .
...	Other parameters passed to the appropriate table function

**Value**

A volker tibble

**Examples**

```
library(volker)
data <- volker:::chatgpt

tab_counts(data, sd_gender)
```

---

tab\_metrics

*Output a table with distribution parameters*


---

**Description**

The table type depends on the number of selected columns:

- One column: see [tab\\_metrics\\_one](#)
- Multiple columns: see [tab\\_metrics\\_items](#)
- One column and one grouping column: see [tab\\_metrics\\_one\\_grouped](#)
- Multiple columns and one grouping column: see [tab\\_metrics\\_items\\_grouped](#)

**Usage**

```
tab_metrics(data, cols, col_group = NULL, clean = TRUE, ...)
```

**Arguments**

data	A data frame
cols	A tidy column selection, e.g. a single column (without quotes) or multiple columns selected by methods such as <code>starts_with()</code> .
col_group	Optional, a grouping column (without quotes).
clean	Prepare data by <a href="#">data_clean</a> .
...	Other parameters passed to the appropriate table function

**Value**

A volker tibble

**Examples**

```
library(volker)
data <- volker::chatgpt

tab_metrics(data, sd_age)
```

# Index

## \* datasets

chatgpt, 2

chatgpt, 2

codebook, 3, 6

data\_clean, 3, 5, 9, 11–13

data\_clean\_sosci, 4

get\_direction, 10, 12

html\_report, 4

idx\_add, 5

labs\_apply, 5

labs\_clear, 6

labs\_restore, 7, 8

labs\_store, 7, 8

plot\_counts, 8, 10

plot\_counts\_items, 8

plot\_counts\_items\_grouped, 8

plot\_counts\_one, 8

plot\_counts\_one\_grouped, 8

plot\_metrics, 9, 11

plot\_metrics\_items, 9

plot\_metrics\_items\_grouped, 9

plot\_metrics\_one, 9

plot\_metrics\_one\_grouped, 9

report\_counts, 10

report\_metrics, 11

tab\_counts, 10, 12

tab\_counts\_items, 12

tab\_counts\_items\_grouped, 12

tab\_counts\_one, 12

tab\_counts\_one\_grouped, 12

tab\_metrics, 11, 13

tab\_metrics\_items, 13

tab\_metrics\_items\_grouped, 13

tab\_metrics\_one, 13

tab\_metrics\_one\_grouped, 13