

# Package ‘overviewR’

May 9, 2026

**Type** Package

**Title** Easily Extracting Information About Your Data

**Version** 0.0.14

**Description** Makes it easy to display descriptive information on a data set. Getting an easy overview of a data set by displaying and visualizing sample information in different tables (e.g., time and scope conditions). The package also provides publishable ‘LaTeX’ code to present the sample information.

**License** GPL-3

**URL** <https://github.com/cosimameyer/overviewR>

**BugReports** <https://github.com/cosimameyer/overviewR/issues>

**Depends** R (>= 3.5.0)

**Imports** data.table (>= 1.14.2), dplyr (>= 1.0.0), ggplot2 (>= 3.3.2),  
ggrepel (>= 0.8.2), ggvenn (>= 0.1.8), rlang, tibble (>= 3.0.1), tidyr

**Suggests** countrycode, covr, devtools, eulerr, knitr, magrittr,  
pkgdown, rmarkdown, spelling, testthat, xtable

**VignetteBuilder** knitr, rmarkdown

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Cosima Meyer [cre, aut],  
Dennis Hammerschmidt [aut]

**Maintainer** Cosima Meyer <cosima.meyer@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-05-05 06:50:02 UTC

## Contents

<i>.overview_heat</i>	2
<i>.overview_tab</i>	3
<i>calculate_share_non_row_wise</i>	4
<i>calculate_share_row_wise</i>	4
<i>find_int_runs</i>	5
<i>overview_add_na_output</i>	5
<i>overview_crossplot</i>	6
<i>overview_crosstab</i>	7
<i>overview_heat</i>	8
<i>overview_latex</i>	9
<i>overview_markdown</i>	11
<i>overview_na</i>	12
<i>overview_overlap</i>	13
<i>overview_plot</i>	14
<i>overview_plot_absolute</i>	15
<i>overview_plot_percentage</i>	16
<i>overview_tab</i>	17
<i>overview_tab_df</i>	18
<i>overview_tab_dt</i>	18
<i>theme_heat_plot</i>	19
<i>theme_na_plot</i>	19
<i>toydata</i>	20
<b>Index</b>	<b>21</b>

---

<i>.overview_heat</i>	<i>.overview_tab</i>
-----------------------	----------------------

---

## Description

Internal function that calculates the ‘*overview\_tab*’ for *data.table* objects

## Usage

```
.overview_heat(
  dat = NULL,
  id = NULL,
  time = NULL,
  label = FALSE,
  perc = FALSE,
  col_low = NULL,
  col_high = NULL,
  xaxis = NULL,
  yaxis = NULL,
  theme_plot = NULL,
  exp_total = NULL,
```

```

  col_names = NULL
)

```

**Arguments**

<code>dat</code>	The data set
<code>id</code>	The scope (e.g., country codes or individual IDs). The axis is ordered in ascending order by default.
<code>time</code>	The time (e.g., time periods given by years, months, ...)
<code>label</code>	If TRUE (default), the total number of observations/percentages of observations are displayed. If FALSE, it returns no labels.
<code>perc</code>	If FALSE (default) plot returns the total number of observations per time-scope-unit. If TRUE, it returns the number of observations per time-scope-unit in percentage
<code>col_low</code>	Hex color code for the lowest value (default is "#dceaf2")
<code>col_high</code>	Hex color code for the highest value (default is "#2A5773")
<code>xaxis</code>	Label of your x axis ("Time frame" is default)
<code>yaxis</code>	Label of your y axis ("Sample" is default)
<code>theme_plot</code>	Previously generated theme
<code>exp_total</code>	Expected total number of observations (i.e. maximum) for time unit.
<code>col_names</code>	The column names (containing id and time)

**Value**

A `ggplot`

---

<code>.overview_tab</code>	<code>.overview_tab</code>
----------------------------	----------------------------

---

**Description**

Internal function that calculates the `'overview_tab'` for `data.table` objects

**Usage**

```
.overview_tab(dat = NULL, id = NULL, time = NULL, col_names = NULL)
```

**Arguments**

<code>dat</code>	Your data set
<code>id</code>	Scope (e.g., country codes or individual IDs)
<code>time</code>	Time (e.g., time periods given by years, months, ...). There are three options to add a date variable: 1) Time can be a character vector containing <b>one</b> time variable, 2) a time variable following the YYYY-MM-DD format, or 3) or a list containing multiple time variables ( <code>'time = list(year = NULL, month = NULL, day = NULL)'</code> ).
<code>col_names</code>	The column names (containing id and time)

**Value**

A data.table

---

```
calculate_share_non_row_wise  
    calculate_share_non_row_wise
```

---

**Description**

Function used in 'overview\_na' to calculate the column-wise share of NA

**Usage**

```
calculate_share_non_row_wise(dat = NULL)
```

**Arguments**

dat                    Data frame

**Value**

The function returns a data set that has the information on the column-wise NA share

---

```
calculate_share_row_wise  
    calculate_share_row_wise
```

---

**Description**

Function used in 'overview\_na' to calculate the share of NA row-wise

**Usage**

```
calculate_share_row_wise(dat = NULL)
```

**Arguments**

dat                    Data frame

**Value**

The function returns a data set that has the information on the row-wise NA share

---

find_int_runs	<i>find_int_runs</i>
---------------	----------------------

---

**Description**

Function used in 'overview\_tab' to find running integers

**Usage**

```
find_int_runs(run = NULL)
```

**Arguments**

run                    Variable (integer) that should be checked for consecutive values

**Value**

The function returns a data set

---

overview_add_na_output	<i>overview_add_na_output</i>
------------------------	-------------------------------

---

**Description**

Function used in 'overview\_na' to generate a new data frame with na\_count and percentage share of NAs for each row

**Usage**

```
overview_add_na_output(dat_result = NULL, dat = NULL)
```

**Arguments**

dat\_result            Data.frame from 'overview\_na'  
dat                    Data frame

**Value**

The function returns a data set that has the information on the row-wise NA share

---

overview\_crossplot      *overview\_crossplot*

---

## Description

This function plots a ggplot to visualize a cross table plot.

## Usage

```
overview_crossplot(  
  dat,  
  id,  
  time,  
  cond1,  
  cond2,  
  threshold1,  
  threshold2,  
  xaxis = "Condition 1",  
  yaxis = "Condition 2",  
  label = FALSE,  
  color = FALSE,  
  dot_size = 2,  
  fontsize = 2.5  
)
```

## Arguments

dat	Your data set
id	Your scope (e.g., country codes or individual IDs). If the id variable contains NAs, they will not be included in the plot.
time	Your time (e.g., time periods given by years, months, ...)
cond1	Variable that describes the first condition
cond2	Variable that describes the second condition
threshold1	A threshold for cond1
threshold2	A threshold for cond2
xaxis	Label of the x axis ("Condition 1" is default)
yaxis	Label of the y axis ("Condition 2" is default)
label	Label of the observations. Overlapping labels are avoided by using 'ggrepel'
color	Color of the different observation groups
dot_size	Option argument that defines the dot size (default is 2)
fontsize	If label is TRUE, the fontsize arguments allows to define the text of the labels (the default is 2.5)

**Value**

A ggplot figure that presents the sample information visually in a cross table

**Examples**

```
data(toydata)
overview_crossplot(
  dat = toydata,
  cond1 = gdp,
  cond2 = population,
  threshold1 = 25000,
  threshold2 = 27000,
  id = ccode,
  time = year
)
```

---

overview_crosstab	<i>overview_crosstab</i>
-------------------	--------------------------

---

**Description**

Sorts a data set conditionally in a cross table. This can be helpful to get a sense of the time and scope conditions of a data set. Note, if used with a data set that has multiple observations on the id-time unit, the function automatically aggregates this information using the mean.

**Usage**

```
overview_crosstab(dat, cond1, cond2, threshold1, threshold2, id, time)
```

**Arguments**

dat	A data set object
cond1	Variable that describes the first condition
cond2	Variable that describes the second condition
threshold1	A threshold for cond1
threshold2	A threshold for cond2
id	Scope (e.g., country codes or individual IDs)
time	Time (e.g., time periods given by years, months, ...)

**Value**

A data frame object that contains a summary of the data set that can later be converted to a 'LaTeX' output using `overview_latex`

**Examples**

```

data(toydata)
overview_crosstab(
  dat = toydata,
  cond1 = gdp,
  cond2 = population,
  threshold1 = 25000,
  threshold2 = 27000,
  id = ccode,
  time = year
)

```

---

 overview\_heat

*overview\_heat*


---

**Description**

This function plots a heat map to visualize the coverage of the time-scope-units of the data. Options include total number of cases per time-scope-unit or relative number in percentage.

**Usage**

```

overview_heat(
  dat,
  id,
  time,
  perc = FALSE,
  exp_total = NULL,
  xaxis = "Time frame",
  yaxis = "Sample",
  col_low = "#dceaf2",
  col_high = "#2A5773",
  label = TRUE
)

```

**Arguments**

<code>dat</code>	The data set
<code>id</code>	The scope (e.g., country codes or individual IDs). The axis is ordered in ascending order by default.
<code>time</code>	The time (e.g., time periods given by years, months, ...)
<code>perc</code>	If FALSE (default) plot returns the total number of observations per time-scope-unit. If TRUE, it returns the number of observations per time-scope-unit in percentage
<code>exp_total</code>	Expected total number of observations (i.e. maximum) for time unit.
<code>xaxis</code>	Label of your x axis ("Time frame" is default)

yaxis	Label of your y axis ("Sample" is default)
col_low	Hex color code for the lowest value (default is "#dceaf2")
col_high	Hex color code for the lowest value (default is "#2A5773")
label	If TRUE (default), the total number of observations/percentages of observations are displayed. If FALSE, it returns no labels.

**Value**

A ggplot figure that presents sample coverage visually

**Examples**

```
data(toydata)
overview_heat(toydata, ccode, year, perc = TRUE, exp_total = 12)
```

---

overview_latex	<i>overview_latex</i>
----------------	-----------------------

---

**Description**

Produces a 'LaTeX' output for output obtained via overview\_tab and overview\_crosstab

**Usage**

```
overview_latex(
  obj,
  title = "Time and scope of the sample",
  id = "Sample",
  time = "Time frame",
  crosstab = FALSE,
  cond1 = "Condition 1",
  cond2 = "Condition 2",
  save_out = FALSE,
  file_path,
  label = "tab:tab1",
  fontsize,
  file,
  path
)
```

**Arguments**

obj	Overview object produced by overview_tab or overview_crosstab
title	Caption of the table (default is "Time and scope of the sample")
id	The name of the left column (default is "Sample"), will be ignored if crosstab is TRUE

time	The name of the right column (default is ("Time frame")), will be ignored if crosstab is TRUE
crosstab	Logical argument, if TRUE produces a crosstab output, default is FALSE
cond1	Description for the first condition (character), will be ignored if crosstab is FALSE. This should correspond to the input for cond1 in overview_crosstab
cond2	Description for the second condition (character), will be ignored if crosstab is FALSE. This should correspond to the input for cond2 in overview_crosstab
save_out	Optional argument, exports the output table as a .tex file, default is FALSE
file_path	Specifies the path and file name (.tex) where you store your output
label	Specifies the label (default is "tab:tab1")
fontsize	Specifies the font size (all 'LaTeX' font sizes such as "scriptsize" or "small" work)
file	This argument is deprecated. Please use "file_path" instead and add the full path.
path	This argument is deprecated. Please use "file_path" instead and add the full path.

### Value

A 'LaTeX' output that can either be copy-pasted in a text document or exported directed as a .tex file

### Examples

```
data(toydata)

overview_object <- overview_tab(dat = toydata, id = ccode, time = year)
overview_latex(
  obj = overview_object,
  title = "Some nice title",
  crosstab = FALSE
)

#' overview_object <- overview_tab(dat = toydata, id = ccode, time = year)
overview_latex(
  obj = overview_object,
  title = "Some nice title",
  file_path = "some/path_to/your_output_file.tex"
)

overview_ct_object <- overview_crosstab(
  dat = toydata,
  cond1 = gdp,
  cond2 = population,
  threshold1 = 25000,
  threshold2 = 27000,
  id = ccode,
  time = year
)
overview_latex(
```

```

obj = overview_ct_object,
title = "Some nice title for a cross tab",
crosstab = TRUE,
cond1 = "Name of first condition",
cond2 = "Name of second condition"
)

```

---

overview\_markdown      *overview\_markdown*

---

## Description

Produces a Markdown table output for objects obtained via `overview_tab` and `overview_crosstab`

## Usage

```

overview_markdown(
  obj,
  title = "Time and scope of the sample",
  id = "Sample",
  time = "Time frame",
  crosstab = FALSE,
  cond1 = "Condition 1",
  cond2 = "Condition 2",
  save_out = FALSE,
  file_path = ""
)

```

## Arguments

<code>obj</code>	Overview object produced by <code>overview_tab</code> or <code>overview_crosstab</code>
<code>title</code>	Optional title printed as a Markdown heading above the table. Defaults to "Time and scope of the sample".
<code>id</code>	The header of the left column (default is "Sample"). Ignored when <code>crosstab = TRUE</code> .
<code>time</code>	The header of the right column (default is "Time frame"). Ignored when <code>crosstab = TRUE</code> .
<code>crosstab</code>	Logical. If <code>TRUE</code> produces a cross-table output. Default is <code>FALSE</code> .
<code>cond1</code>	Label for the first condition. Ignored when <code>crosstab = FALSE</code> .
<code>cond2</code>	Label for the second condition. Ignored when <code>crosstab = FALSE</code> .
<code>save_out</code>	Logical. If <code>TRUE</code> writes the output to a file. Default is <code>FALSE</code> .
<code>file_path</code>	Path (including file name with <code>.md</code> extension) where the output is stored when <code>save_out = TRUE</code> .

**Value**

Invisibly returns a character string with the Markdown table. The table is also printed to the console (unless `save_out = TRUE`).

**Examples**

```
data(toydata)

overview_object <- overview_tab(dat = toydata, id = ccode, time = year)
overview_markdown(obj = overview_object)

overview_ct_object <- overview_crosstab(
  dat = toydata,
  cond1 = gdp,
  cond2 = population,
  threshold1 = 25000,
  threshold2 = 27000,
  id = ccode,
  time = year
)
overview_markdown(
  obj = overview_ct_object,
  crosstab = TRUE,
  cond1 = "GDP",
  cond2 = "Population"
)
```

---

`overview_na`*overview\_na*

---

**Description**

This function plots a ggplot to visualize the distribution of NAs across all variables in the data set.

**Usage**

```
overview_na(
  dat,
  yaxis = "Variables",
  perc = TRUE,
  row_wise = FALSE,
  add = FALSE
)
```

**Arguments**

<code>dat</code>	Your data set
<code>yaxis</code>	Label of your y axis ("Variables" is default)

perc	If TRUE (default) plot returns the number of NAs in percentage
row_wise	If TRUE (FALSE is default) plot return the number of NAs per row
add	If TRUE (FALSE is default) it generates a new data frame with na_count and percentage share of NAs for each row

**Value**

Depending on the selection, the function returns a ggplot figure that presents the distribution of NAs in the data set or adds the information on the row-wise NA share

**Examples**

```
data(toydata)
overview_na(toydata, perc = FALSE)
```

---

overview_overlap	<i>overview_overlap</i>
------------------	-------------------------

---

**Description**

Provides an overview of the overlap of two data sets. Cautionary note: This function is currently only preliminary workable and can only capture 2 data sets. We are working on an extension that allows to compare multiple data sets.

**Usage**

```
overview_overlap(
  dat1,
  dat2,
  dat1_id,
  dat2_id,
  dat1_name = "Data set 1",
  dat2_name = "Data set 2",
  plot_type = "bar",
  proportional = FALSE
)
```

**Arguments**

dat1	A first data set object
dat2	A second data set object
dat1_id	Scope (e.g., country codes or individual IDs) of dat1. It is important that both ID variables are exactly the same to generate the perfect match.
dat2_id	Scope (e.g., country codes or individual IDs) of dat2. It is important that both ID variables are exactly the same to generate the perfect match.
dat1_name	Name of dat1 ("Data set 1" is the default)

dat2_name	Name of dat2 ("Data set 2" is the default)
plot_type	Type of plot ("bar" and "venn" are the two options) "venn" relies on the ggvenn function
proportional	Logical. If TRUE and plot_type = "venn", draws a proportional Euler diagram using the <b>eulerr</b> package (must be installed separately). The returned object is an eulergram (base-R graphics), not a ggplot2 object. Default is FALSE.

### Value

A ggplot2 object (bar chart or standard Venn diagram), or an eulergram object when proportional = TRUE.

### Examples

```
## Not run:
data(toydata)
toydata2 <- toydata[which(toydata$year > 1992), ]
overview_overlap(
  dat1 = toydata, dat2 = toydata2, dat1_id = ccode,
  dat2_id = ccode
)

## End(Not run)
```

---

overview\_plot

*overview\_plot*

---

### Description

This function plots a ggplot to visualize the distribution of scope objects across the time frame.

### Usage

```
overview_plot(
  dat,
  id,
  time,
  xaxis = "Time frame",
  yaxis = "Sample",
  asc = TRUE,
  color,
  dot_size = 2
)
```

**Arguments**

<code>dat</code>	Your data set
<code>id</code>	Your scope (e.g., country codes or individual IDs). If the <code>id</code> variable contains NAs, they will not be included in the plot.
<code>time</code>	Your time (e.g., time periods given by years, months, ...)
<code>xaxis</code>	Label of the x axis ("Time frame" is default)
<code>yaxis</code>	Label of the y axis ("Sample" is default)
<code>asc</code>	Sorting the y axis in ascending order ("TRUE" is default)
<code>color</code>	Optional argument. Either a column variable (for conditional coloring) or a color string (e.g., "steelblue") to paint all lines/points a fixed color. When <code>color</code> is a column variable the returned <code>ggplot</code> object can be extended with <code>+ ggplot2::scale_color_manual()</code> or <code>+ ggplot2::scale_color_brewer()</code> to apply custom palettes.
<code>dot_size</code>	Option argument that defines the dot size (default is 2)

**Value**

A `ggplot` figure that presents the sample information visually

**Examples**

```
data(toydata)
overview_plot(dat = toydata, id = ccode, time = year)
```

---

overview\_plot\_absolute  
*overview\_plot\_absolute*

---

**Description**

Function used in 'overview\_na' to plot the absolute share of NA values

**Usage**

```
overview_plot_absolute(  
  dat_result = NULL,  
  theme_plot = NULL,  
  yaxis = NULL,  
  xaxis = NULL  
)
```

**Arguments**

<code>dat_result</code>	Data frame
<code>theme_plot</code>	Theme for the plot (pre-defined)
<code>yaxis</code>	Name for yaxis
<code>xaxis</code>	Name for xaxis

**Value**

The function returns a ggplot

---

`overview_plot_percentage`  
*overview\_plot\_percentage*

---

**Description**

Function used in 'overview\_na' to plot the percentage share of NA values

**Usage**

```
overview_plot_percentage(  
  dat_result = NULL,  
  theme_plot = NULL,  
  yaxis = NULL,  
  xaxis = NULL  
)
```

**Arguments**

<code>dat_result</code>	Data frame
<code>theme_plot</code>	Theme for the plot (pre-defined)
<code>yaxis</code>	Name for yaxis
<code>xaxis</code>	Name for xaxis

**Value**

The function returns a ggplot

---

overview_tab	<i>overview_tab</i>
--------------	---------------------

---

### Description

Provides an overview table for the time and scope conditions of a data set. If a `data.table` object is provided, the function uses `data.table`'s syntax to perform the evaluation

### Usage

```
overview_tab(
  dat,
  id,
  time = list(year = NULL, month = NULL, day = NULL),
  complex_date = FALSE
)
```

### Arguments

dat	A data frame or data table object
id	Scope (e.g., country codes or individual IDs)
time	Time (e.g., time periods given by years, months, ...). There are three options to add a date variable: 1) Time can be a character vector containing <b>one</b> time variable, 2) a time variable following the YYYY-MM-DD format, or 3) or a list containing multiple time variables ('time = list(year = NULL, month = NULL, day = NULL)').
complex_date	Boolean argument identifying if there is a more complex (list-wise) date_time parameter (FALSE is the default)

### Value

A data frame object that contains a summary of a sample that can later be converted to a 'LaTeX' output using `overview_latex`

### Examples

```
# With version 1 (and also 2):

data(toydata)
output_table <- overview_tab(dat = toydata, id = ccode, time = year)

# With version 3:
overview_tab(dat = toydata, id = ccode, time = list(
  year = toydata$year,
  month = toydata$month, day = toydata$day
), complex_date = TRUE)
```

---

overview_tab_df	<i>overview_tab_df</i>
-----------------	------------------------

---

**Description**

Internal function that calculates the ‘overview\_tab’ for data.frame objects

**Usage**

```
overview_tab_df(dat2 = NULL, dat = NULL, id = NULL, time = NULL)
```

**Arguments**

dat2	Your data set
dat	Your data set
id	Scope (e.g., country codes or individual IDs)
time	Time (e.g., time periods given by years, months, ...). There are three options to add a date variable: 1) Time can be a character vector containing <b>one</b> time variable, 2) a time variable following the YYYY-MM-DD format, or 3) or a list containing multiple time variables (*time = list(year = NULL, month = NULL, day = NULL)*).

**Value**

A data.frame

---

overview_tab_dt	<i>overview_tab_dt</i>
-----------------	------------------------

---

**Description**

Internal function that calculates the ‘overview\_tab’ for data.table objects

**Usage**

```
overview_tab_dt(dat = NULL, id = NULL, time = NULL, col_names = NULL)
```

**Arguments**

dat	Your data set
id	Scope (e.g., country codes or individual IDs)
time	Time (e.g., time periods given by years, months, ...). There are three options to add a date variable: 1) Time can be a character vector containing <b>one</b> time variable, 2) a time variable following the YYYY-MM-DD format, or 3) or a list containing multiple time variables (*time = list(year = NULL, month = NULL, day = NULL)*).
col_names	The column names (containing id and time)

**Value**

A `data.table`

---

<code>theme_heat_plot</code>	<i>theme_heat_plot</i>
------------------------------	------------------------

---

**Description**

Defines the theme for the ‘`overview_heat`’ plot function

**Usage**

```
theme_heat_plot()
```

**Value**

A theme for the ‘`overview_heat`’ plot

---

<code>theme_na_plot</code>	<i>theme_na_plot</i>
----------------------------	----------------------

---

**Description**

Defines the theme for the ‘`overview_na`’ plot function

**Usage**

```
theme_na_plot()
```

**Value**

A theme for the ‘`overview_na`’ plot

---

toydata

*Cross-sectional data for countries*

---

### Description

Small, artificially generated toy data set that comes in a cross-sectional format where the unit of analysis is either country-year or country-year-month. It provides artificial information for five countries (Angola, Benin, France, Rwanda, and the UK) for a time span from 1990 to 1999 to illustrate the use of the package.

### Usage

```
data(toydata)
```

### Format

An object of class "data.frame"

**ccode** ISO3 country code (as character) for the countries in the sample (Angola, Benin, France, Rwanda, and UK)

**year** A value between 1990 and 1999

**month** An abbreviation (MMM) for month (character)

**gpd** A fake value for GDP (randomly generated)

**population** A fake value for population (randomly generated)

### References

This data set was artificially created for the overviewR package.

### Examples

```
data(toydata)
head(toydata)
```

# Index

## \* datasets

- toydata, [20](#)
- .overview\_heat, [2](#)
- .overview\_tab, [3](#)
  
- calculate\_share\_non\_row\_wise, [4](#)
- calculate\_share\_row\_wise, [4](#)
  
- find\_int\_runs, [5](#)
  
- overview\_add\_na\_output, [5](#)
- overview\_crossplot, [6](#)
- overview\_crosstab, [7](#)
- overview\_heat, [8](#)
- overview\_latex, [9](#)
- overview\_markdown, [11](#)
- overview\_na, [12](#)
- overview\_overlap, [13](#)
- overview\_plot, [14](#)
- overview\_plot\_absolute, [15](#)
- overview\_plot\_percentage, [16](#)
- overview\_tab, [17](#)
- overview\_tab\_df, [18](#)
- overview\_tab\_dt, [18](#)
  
- theme\_heat\_plot, [19](#)
- theme\_na\_plot, [19](#)
- toydata, [20](#)