

# Package: mtb (via r-universe)

August 26, 2024

**Type** Package

**Title** My Toolbox for Assisting Document Editing and Data Presenting

**Version** 0.1.8

**Description** The purpose of this package is to share a collection of functions the author wrote during weekends for managing kitchen and garden tasks, e.g. making plant growth charts or Thanksgiving kitchen schedule charts, etc. Functions might include but not limited to: (1) aiding summarizing time related data; (2) generating axis transformation from data; and (3) aiding Markdown (with html output) and Shiny file editing.

**License** AGPL (>= 3)

**Encoding** UTF-8

**Depends** R (>= 4.1)

**Imports** htmltools (>= 0.4.0), ggplot2 (>= 3.3.0), scales (>= 1.0.0), data.table (>= 1.14.2), labeling (>= 0.3)

**Suggests** rmarkdown (>= 1.16), knitr, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**URL** <https://github.com/yh202109/mtb>

**RoxygenNote** 7.2.1

**Language** en-US

**LazyData** true

**Repository** <https://yh202109.r-universe.dev>

**RemoteUrl** <https://github.com/yh202109/mtb>

**RemoteRef** HEAD

**RemoteSha** 1444ff3f63052c0e10070a94c4046d3c8608a8e6

## Contents

add_colored_box	2
add_colored_str	3
bill_cross_check	4
bill_cross_count	5
color_set_palette	6
color_test_palette	7
exdt	7
time_plot_event	8
time_plot_interval	9
trans_composition	10
trans_loglinear	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

add_colored_box	<i>Add a box with specified color in an R Markdown file.</i>
-----------------	--

---

### Description

add\_colored\_box returns a box component generated by htmltools with specified color and styles.

### Usage

```
add_colored_box(
  type = "blue-default",
  label = "",
  info = "place details here using info option",
  bgcolor = NULL,
  width = 0.5,
  halign = "c",
  top = FALSE
)
```

### Arguments

type	One of: <ul style="list-style-type: none"> <li>• NULL for no default color or label</li> <li>• 'blue-default' for a steel-blue box</li> <li>• 'gray-info' for a gray box</li> <li>• 'blue-info' for a blue box</li> <li>• 'green-remainder' for a green box</li> <li>• 'yellow-warning' for a yellow box</li> <li>• 'red-stop' for a red box</li> </ul>
label	One of:

	<ul style="list-style-type: none"> <li>• NULL for no label if type is NULL or using label set by type</li> <li>• A string shown on the top of box</li> </ul>
info	A string including the main message of the box
bgcolor	NA or a length 3 vector with integer elements between 0 to 255
width	NA or a number between 0.25 to 0.95
halign	One of: <ul style="list-style-type: none"> <li>• NA for center aligned</li> <li>• 'c' for center aligned</li> <li>• 'r' for right aligned</li> </ul>
top	One of: <ul style="list-style-type: none"> <li>• NA</li> <li>• FALSE for inline</li> <li>• TRUE for top-of-page</li> </ul>

### Examples

```
add_colored_box( type='blue-default', info='the document include information regarding...')
```

---

add\_colored\_str      *Add a string with specified color or background color.*

---

### Description

add\_colored\_str returns a string component generated by htmltools with specified color and styles.

### Usage

```
add_colored_str(
  text = "",
  color = c(51, 122, 183),
  alpha = 255,
  bgcolor = NULL,
  bgamma = 51,
  fontsize = 1,
  bold = FALSE,
  it = FALSE
)
```

**Arguments**

text	A string. default="".
color	One of <ul style="list-style-type: none"> <li>• a color name, e.g. 'red'.</li> <li>• a HEX color string, e.g. '#000000' or '#000000FF'.</li> <li>• an RGB vector for the color of text</li> </ul>
alpha	An integer between 1 and 255 for text alpha. default=255.
bgcolor	One of <ul style="list-style-type: none"> <li>• a color name, e.g. 'red'.</li> <li>• a HEX color string, e.g. '#000000' or '#000000FF'.</li> <li>• an RGB vector for the color of text</li> </ul>
bgalpha	An integer between 1 and 255 for background alpha. default=51.
fontsize	A real number between 0.5 and 5.0 for font size. default=1.
bold	A logical value for bold fonts. default=FALSE.
it	A Boolean value for italic fonts. default=FALSE.

**Value**

a formatted string

**Examples**

```
add_colored_str("warning: read this message carefully.", color = c(255, 0, 0))
```

---

bill_cross_check	<i>Check two tables with unique matching ids and generate reports on duplicated ids and repeated columns when applicable..</i>
------------------	--

---

**Description**

bill\_cross\_check returns a merged data.table showing information regarding ids and repeated columns.

**Usage**

```
bill_cross_check(dt1 = NULL, dt2 = NULL, id = NULL, chk = NULL)
```

**Arguments**

dt1	A table.
dt2	A table.
id	A column name or a vector of column names
chk	A column name

**Value**

a data.table

**Examples**

```
bill_cross_check(data.frame(col1=c(1,2,3,3), col2=c('a','b','c','c'),
col3=c('-', '=', '+', '-')), data.frame(col1=c(1,2,3), col2=c('a','b','c'),
col3=c('-', '=', '+')), id=c('col1', 'col2'), chk='col3')
```

---

bill_cross_count	<i>Summarize times being purchased for individual items' from multiple grocery shopping lists.</i>
------------------	--

---

**Description**

bill\_cross\_count returns a data.table showing how many times each items being listed on individual bills.

**Usage**

```
bill_cross_count(
  ldt = list(),
  id = NULL,
  gp = NULL,
  type = "count",
  condstr = ""
)
```

**Arguments**

ldt	A list of grocery tables in data.frame format.
id	A column name
gp	A column name or a vector of column names
type	A string in one of the following: <ul style="list-style-type: none"> <li>• count for number of rows in each bill</li> <li>• cond for conditional counting</li> <li>• condwt for conditional counting with total in parenthesis</li> </ul>
condstr	A string for conditional counting

**Value**

a data.table

**Examples**

```
bill_cross_count(list(cbind(col1=c('a','b','c'),col2=c(1,2,3)),
  cbind(col1=c('d','c','d'),col2=c(4,5,6))), id='col1')
```

---

color\_set\_palette      *Generate a color vector*

---

**Description**

Create a list of colors for a data vector by a list major colors.

**Usage**

```
color_set_palette(
  vect = c(),
  vectn = c(),
  cols = c("blue", "cyan", "darkorange"),
  black = "",
  gray9 = ""
)
```

**Arguments**

vect	A vector for groups.
vectn	An integer vector with length 0 or with the same length of vect for order of elements in vect. default=c()
cols	One of <ul style="list-style-type: none"> <li>• A color names vector</li> <li>• An RGB triplet vector</li> <li>• A HEX vector</li> </ul>
black	A level in vect that should be assigned to black color. default=""
gray9	A level in vect that should be assigned to gray9 color. default=""

**Value**

a named vector

**Examples**

```
color_set_palette( c('apple', 'orange', 'lime', 'apple'), c(2,1,3,2), 'red', 'blue')
```

---

color\_test\_palette      *Test a color vector*

---

**Description**

Create a figure using the assigned color vector

**Usage**

```
color_test_palette(colvect = c(), type = "line")
```

**Arguments**

colvect	A vector returned by color_set_palette()
type	One of <ul style="list-style-type: none"><li>• 'line' for using the color vector on a line plot (Default)</li><li>• 'box' for using the color vector on a box plot.</li></ul>

**Value**

A plot

**Examples**

```
color_test_palette( setNames(c(1,2,3,4), c('apple','orange','avocado','lime') ))
```

---

exdt                      *Example dataset*

---

**Description**

Example dataset

**Author(s)**

package author

---

time_plot_event	<i>Plot labeled events</i>
-----------------	----------------------------

---

### Description

Create a plot for events with labels

### Usage

```
time_plot_event(dt, xlab = "Time", anchor = TRUE, compact = FALSE)
```

### Arguments

dt	a data.frame with the following columns <ul style="list-style-type: none"> <li>• idfor ID of each group</li> <li>• idnfor order of ID</li> <li>• startfor starting time</li> <li>• endfor ending time with arrow head</li> <li>• labelfor labeling the starting time</li> <li>• labelendfor labeling the ending time of a interval</li> <li>• typefor event type as one of p (point), i (interval), b (box)</li> <li>• colora string for event color</li> </ul>
xlab	A string for the x-axis title
anchor	A Boolean value for the vertical lines linking start to the x-axis
compact	A Boolean value for reducing the vertical spacing when applicable

### Value

a plot

### Examples

```
library(ggplot2)
dt = data.frame( id=paste('member',c(rep(c(1,2,3),each=3),3),sep=""),
  idn=c(rep(1,3),rep(-1,3), rep(2,4)),
  start=1800*c(0,1,2, 0.5, 1.2, 3, 1,2,3,4),
  end=1800*c(2,NA,3, 2, 6, NA, 2,2.5,3, 3.5),
  label=c(paste('event-',seq(1,10),sep='')),
  labelend=c('','?',')','>','X'),
  type=c('b', 'p', 'i','i','p','p','p','b','i','i' ),
  color=c('stove', 'oven', 'oven','oven','stove','oven','oven','other','stove','oven' )
)
time_plot_event( dt )
```



---

time\_plot\_interval      *Plot periods of events*

---

## Description

Create a plot for event periods by ID

## Usage

```
time_plot_interval(  
  dt,  
  xlab = "DateTime",  
  ylab = "ID",  
  legend_title = "Group",  
  arrow_wt = 1,  
  arrow_color = "black"  
)
```

## Arguments

dt	a data.frame with the following columns <ul style="list-style-type: none"><li>• id for ID of each interval</li><li>• idn for order of ID</li><li>• start for starting time</li><li>• end for ending time with arrow head</li><li>• label for labeling the starting time</li></ul>
xlab	A string for the label of X-axis
ylab	A string for the label of Y-axis
legend_title	A string for the title of legend
arrow_wt	An integer for the weight of arrow
arrow_color	A string for the color of arrow

## Value

a plot

## Examples

```
library(ggplot2)  
dt = data.frame( id=c('ID01', 'ID12', 'ID3'), idn=c(1,3,2), start=1800*c(0,1,2), end=1800*c(2,-1,3),  
  label=c('A', 'B', 'C') )  
time_plot_interval( dt, xlab='Time', ylab='ID', legend_title='Group', arrow_wt=3,  
  arrow_color='gray')
```

---

trans_composition	<i>Transformation for continuous data with a finite number of distinct values</i>
-------------------	---

---

### Description

trans\_composition() derives a transformation from a numerical vector with a smaller number (ideally < 30) of distinct values. The return can be used with function ggplot::scale\_x\_continuous() or ggplot::scale\_y\_continuous() to create a desired axis.

### Usage

```
trans_composition(x = NULL, nb = 30, brk = NA, dab = NA, dgrd = NA, dgrd2 = NA)
```

### Arguments

x	A numerical vector used in a plot as (typically) x
nb	An integer for the maximum number of breaks. Default=30
brk	One of <ul style="list-style-type: none"> <li>• A numerical value within range(x). All values after the value will be spaced equally</li> <li>• NA or a numerical value that is greater than or equal to max(x). All values will be plotted in the original scale</li> <li>• A numerical value that is smaller than or equal to max(x). All values will be plotted in equal space</li> </ul>
dab	One of <ul style="list-style-type: none"> <li>• NA for a value calculated automatically</li> <li>• A number for the distance after brk</li> </ul>
dgrd	One of <ul style="list-style-type: none"> <li>• NA for a value calculated automatically</li> <li>• A number for the minimum space between major grids</li> </ul>
dgrd2	One of <ul style="list-style-type: none"> <li>• NA for a value calculated automatically</li> <li>• A number for the minimum space between major grids</li> </ul>

### Value

A transformation function

**Examples**

```
library(ggplot2)
pdt=data.frame(x=rep(c(0.5, 1, 10,11,12, 100, 1000), each=5))
pdt$y=pdt$x+rnorm(length(pdt$x))
t=trans_composition(pdt$x,brk=50, dab=3)
ggplot(pdt, aes(x=x, y=y))+geom_point()+scale_x_continuous(trans=t)
```

---

trans_loglinear	<i>Transformation for continuous data with a finite number of distinct values</i>
-----------------	---

---

**Description**

trans\_loglinear() derives a log transformation from a numerical vector with a smaller number (ideally < 30) of distinct values.. The return can be used with function ggplot::scale\_x\_continuous() or ggplot::scale\_y\_continuous() to create a desired axis.

**Usage**

```
trans_loglinear(x = NULL, nb = 30, int = NA, scale = NA, mindist = 0.03)
```

**Arguments**

x	A numerical vector used in a plot as (typically) x
nb	An integer for the maximum number of breaks. Default=30
int	One of <ul style="list-style-type: none"> <li>• NA for a value calculated automatically</li> <li>• A real number (<math>\geq 0</math>) for the shift before log transform</li> </ul>
scale	One of <ul style="list-style-type: none"> <li>• NA for a value calculated automatically</li> <li>• A real number (<math>&gt; 0</math>) for the scale before log transform</li> </ul>
mindist	One of <ul style="list-style-type: none"> <li>• NA for a default value set to 0.03</li> <li>• A real number between 0 and 0.2 for the minimum distance ratio between major ticks</li> </ul>

**Value**

A transformation function

**Examples**

```
library(ggplot2)
pdt=data.frame(x=rep(c(0.5, 1, 10,11,12, 100, 1000), each=5))
pdt$y=pdt$x+rnorm(length(pdt$x))
t=trans_loglinear(pdt$x)
ggplot(pdt, aes(x=x, y=y))+geom_point()+scale_x_continuous(trans=t)
```

# Index

## \* data

- exdt, [7](#)
  
- add\_colored\_box, [2](#)
- add\_colored\_str, [3](#)
  
- bill\_cross\_check, [4](#)
- bill\_cross\_count, [5](#)
  
- color\_set\_palette, [6](#)
- color\_test\_palette, [7](#)
  
- exdt, [7](#)
  
- time\_plot\_event, [8](#)
- time\_plot\_interval, [9](#)
- trans\_composition, [10](#)
- trans\_loglinear, [11](#)