Package 'mschart'

May 16, 2026

Type Package

Title Chart Generation for 'Microsoft Word', 'Microsoft Excel' and 'Microsoft PowerPoint' Documents

Version 0.5.0

Description Create native charts for 'Microsoft PowerPoint', 'Microsoft Excel' and 'Microsoft Word' documents. The resulting charts can then be edited and annotated in the host application. It provides functions to create charts and to modify their content and formatting. The chart's underlying data is automatically saved within the 'Word', 'Excel' or 'PowerPoint' file. It extends the 'officer' package, which does not provide native 'Microsoft' chart production.

License MIT + file LICENSE

URL https://ardata-fr.github.io/officeverse/, https://ardata-fr.github.io/mschart/

BugReports https://github.com/ardata-fr/mschart/issues

Depends R (>= 3.5)

Imports cellranger, data.table, grDevices, htmltools, officer (>= 0.7.5), scales, stats, utils, writexl, xml2 (>= 1.1.0)

Suggests doconv (>= 0.4.0), testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author David Gohel [aut, cre],
ArData [cph],
YouGov [fnd],
Jan Marvin Garbuszus [ctb] (support for openxlsx2 and combo charts),
Stefan Moog [ctb] (support to set chart and plot area color and border, fix for line style none),

Eli Daniels [ctb],
Marlon Molina [ctb] (added table feature),
Rokas Klydzia [ctb] (custom labels),
David Camposeco [ctb] (chart_data_smooth function),
Dan Joplin [ctb] (fix scatter plot data structure)

Maintainer David Gohel <david.gohel@ardata.fr>

Repository CRAN

Date/Publication 2026-05-16 21:50:37 UTC

Contents

`as_bar_stack` *Set a barchart as a stacked barchart*

Description

Apply settings to an `ms_barchart` object to produce a stacked barchart. Options are available to use percentage instead of values and to choose if bars should be vertically or horizontally drawn.

Usage

```
as_bar_stack(x, dir = "vertical", percent = FALSE, gap_width = 50)
```

Arguments

`x`	an `ms_barchart` object
`dir`	the direction of the bars in the chart, value must be one of "horizontal" or "vertical".
`percent`	should bars be displayed as percentages.
`gap_width`	gap width between bars for each category on a bar chart, as a percentage of the bar width. It can be set between 0 and 500.

Value

An `ms_chart` object.

See Also

`chart_settings()`, `ms_barchart()`

Examples

```
library(officer)

my_bar_stack_01 <- ms_barchart(data = browser_data, x = "browser",
  y = "value", group = "serie")
my_bar_stack_01 <- as_bar_stack( my_bar_stack_01 )

my_bar_stack_02 <- ms_barchart(data = browser_data, x = "browser",
  y = "value", group = "serie")
my_bar_stack_02 <- as_bar_stack( my_bar_stack_02, percent = TRUE,
```

```
  dir = "horizontal" )

doc <- read_pptx()
doc <- add_slide(doc, layout = "Title and Content", master = "Office Theme")
doc <- ph_with(doc, my_bar_stack_02, location = ph_location_fullsize())

fileout <- tempfile(fileext = ".pptx")
print(doc, target = fileout)
```

---

body_add_chart *Add a chart to a Word document*

Description

Add a ms_chart to an rdocx object. The graphic will be inserted in an empty paragraph.

Usage

```
body_add_chart(x, chart, style = NULL, pos = "after", width = 5, height = 3)
```

Arguments

x	an rdocx object
chart	an ms_chart object.
style	paragraph style
pos	where to add the new element relative to the cursor, one of "after", "before", "on".
height, width	height and width in inches.

Value

An rdocx object.

See Also

ph_with.ms_chart()

Examples

```
library(officer)
my_barchart <- ms_barchart(data = browser_data,
  x = "browser", y = "value", group = "serie")
my_barchart <- chart_settings( my_barchart, grouping = "stacked",
  gap_width = 50, overlap = 100 )

doc <- read_docx()
doc <- body_add_chart(doc, chart = my_barchart, style = "centered")
print(doc, target = tempfile(fileext = ".docx"))
```

browser_data *Dummy dataset for barchart*

Description

A dataset containing two categorical variables and one integer variable:

Format

A data frame with 18 rows and 3 variables

Details

- browser web browser
- serie id of series
- value integer values

browser_ts *Dummy dataset for time series charts*

Description

A dataset containing a date, a categorical, and an integer variable:

Format

A data frame with 36 rows and 3 variables

Details

- date date values
- browser web browser
- freq values in percent

chart_ax_x *X axis settings*

Description

Define settings for an x axis. S3 generic; the `default` method is documented below. ChartEx charts have a leaner set of supported options - see the package documentation.

Usage

```
chart_ax_x(x, ...)

## Default S3 method:
chart_ax_x(
  x,
  orientation,
  crosses,
  cross_between,
  major_tick_mark,
  minor_tick_mark,
  tick_label_pos,
  display,
  num_fmt,
  rotation,
  limit_min,
  limit_max,
  position,
  major_unit,
  minor_unit,
  major_time_unit,
  minor_time_unit,
  ...
)
```

Arguments

`x`	an `ms_chart` object.
`...`	arguments passed to S3 methods.
`orientation`	axis orientation, one of 'maxMin', 'minMax'.
`crosses`	specifies how the axis crosses the perpendicular axis, one of 'autoZero', 'max', 'min'.
`cross_between`	specifies how the value axis crosses the category axis between categories, one of 'between', 'midCat'.
`major_tick_mark, minor_tick_mark`	tick marks position, one of 'cross', 'in', 'none', 'out'.

tick_label_pos	ticks labels position, one of 'high', 'low', 'nextTo', 'none'.
display	should the axis be displayed (a logical of length 1).
num_fmt	number formatting. See the num_fmt section for more details.
rotation	rotation angle. Value should be between -360 and 360.
limit_min	minimum value on the axis. Date objects are also accepted and will be converted automatically.
limit_max	maximum value on the axis. Date objects are also accepted and will be converted automatically.
position	the value at which this axis crosses the perpendicular axis.
major_unit	numeric, interval between major ticks and gridlines.
minor_unit	numeric, interval between minor ticks and gridlines.
major_time_unit	time unit for major ticks on date axes, one of "days", "months", "years".
minor_time_unit	time unit for minor ticks on date axes, one of "days", "months", "years".

Value

An ms_chart object.

num_fmt

All % need to be doubled, 0%% means "a number and percent symbol".

To my current knowledge, depending on the chart type and options, the following values are not systematically used by office chart engine; i.e. when chart pre-compute percentages, it seems using 0%% will have no effect.

- General: default value
- 0: display the number with no decimal
- 0.00: display the number with two decimals
- 0%%: display as percentages
- 0.00%%: display as percentages with two decimal places
- #,##0
- #,##0.00
- 0.00E+00
- # ?/?
- # ??/??
- mm-dd-yy
- d-mmm-yy
- d-mmm
- mmm-yy
- h:mm AM/PM
- h:mm:ss AM/PM

- h:mm
- h:mm:ss
- m/d/yy h:mm
- #,##0 ;(#,##0)
- #,##0 ;[Red](#,##0)
- #,##0.00;(#,##0.00)
- #,##0.00;[Red](#,##0.00)
- mm:ss
- [h]:mm:ss
- mmss.0
- ##0.0E+0
- @

Illustrations

See Also

chart_ax_y(), ms_areachart(), ms_barchart(), ms_scatterchart(), ms_linechart()

Examples

```
library(mschart)

chart_01 <- ms_linechart(
  data = us_indus_prod,
  x = "date", y = "value",
  group = "type"
)

chart_01 <- chart_ax_y(x = chart_01, limit_min = 20, limit_max = 120)
chart_01

# control axis intervals
chart_01 <- chart_ax_x(chart_01,
  major_unit = 10, major_time_unit = "years"
)
chart_01 <- chart_ax_y(chart_01, major_unit = 20)
chart_01
```

chart_ax_y *Y axis settings*

Description

Define settings for a y axis. S3 generic; the `default` method is documented below. ChartEx charts have a leaner set of supported options - see the package documentation.

Usage

```
chart_ax_y(x, ...)

## Default S3 method:
chart_ax_y(
  x,
  orientation,
  crosses,
  cross_between,
  major_tick_mark,
  minor_tick_mark,
  tick_label_pos,
  display,
  num_fmt,
  rotation,
  limit_min,
  limit_max,
  position,
  major_unit,
  minor_unit,
  major_time_unit,
  minor_time_unit,
  ...
)
```

Arguments

`x`	an `ms_chart` object.
`...`	arguments passed to S3 methods.
`orientation`	axis orientation, one of 'maxMin', 'minMax'.
`crosses`	specifies how the axis crosses the perpendicular axis, one of 'autoZero', 'max', 'min'.
`cross_between`	specifies how the value axis crosses the category axis between categories, one of 'between', 'midCat'.
`major_tick_mark, minor_tick_mark`	tick marks position, one of 'cross', 'in', 'none', 'out'.

tick_label_pos	ticks labels position, one of 'high', 'low', 'nextTo', 'none'.
display	should the axis be displayed (a logical of length 1).
num_fmt	number formatting. See the num_fmt section for more details.
rotation	rotation angle. Value should be between -360 and 360.
limit_min	minimum value on the axis. Date objects are also accepted and will be converted automatically.
limit_max	maximum value on the axis. Date objects are also accepted and will be converted automatically.
position	the value at which this axis crosses the perpendicular axis.
major_unit	numeric, interval between major ticks and gridlines.
minor_unit	numeric, interval between minor ticks and gridlines.
major_time_unit	time unit for major ticks on date axes, one of "days", "months", "years".
minor_time_unit	time unit for minor ticks on date axes, one of "days", "months", "years".

Value

An ms_chart object.

Illustrations

num_fmt

All % need to be doubled, 0%% means "a number and percent symbol".

To my current knowledge, depending on the chart type and options, the following values are not systematically used by office chart engine; i.e. when chart pre-compute percentages, it seems using 0%% will have no effect.

- General: default value
- 0: display the number with no decimal
- 0.00: display the number with two decimals
- 0%%: display as percentages
- 0.00%%: display as percentages with two decimal places
- #,##0
- #,##0.00
- 0.00E+00
- # ?/?
- # ??/??
- mm-dd-yy
- d-mmm-yy
- d-mmm

- mmm-yy
- h:mm AM/PM
- h:mm:ss AM/PM
- h:mm
- h:mm:ss
- m/d/yy h:mm
- #,##0 ;(#,##0)
- #,##0 ;[Red](#,##0)
- #,##0.00;(#,##0.00)
- #,##0.00;[Red](#,##0.00)
- mm:ss
- [h]:mm:ss
- mmss.0
- ##0.0E+0
- @

See Also

chart_ax_x(), ms_areachart(), ms_barchart(), ms_scatterchart(), ms_linechart()

Examples

```
library(officer)
library(mschart)

chart_01 <- ms_linechart(
  data = us_indus_prod,
  x = "date", y = "value",
  group = "type"
)
chart_01 <- chart_settings(chart_01, style = "marker")
chart_01 <- chart_ax_x(
  x = chart_01, num_fmt = "[$-fr-FR]mmm yyyy",
  limit_min = min(us_indus_prod$date),
  limit_max = as.Date("1992-01-01")
)
chart_01
```

---

chart_data_fill *Modify fill colour*

Description

Specify mappings from levels in the data to displayed fill colours.

Usage

```
chart_data_fill(x, values, update_stroke = TRUE)
```

Arguments

x	an ms_chart object.
values	character(num of series|1): a set of colour values to map data values to. It is a named vector, the values will be matched based on the names. If it contains only one colour, this colour will be associated to all existing series.
update_stroke	if TRUE (the default), the series stroke colour is also updated to values, so that the filled shape and its border match. Series whose stroke was set to "transparent" (for example by the ms_areachart() and ms_piechart() constructors) are left untouched to preserve that deliberate "no border" default. Set to FALSE to keep the current stroke colours untouched and manage them independently via chart_data_stroke().

Value

An ms_chart object.

See Also

Other Series customization functions: chart_data_line_style(), chart_data_line_width(), chart_data_size(), chart_data_smooth(), chart_data_stroke(), chart_data_symbol(), chart_labels_text()

Examples

```
my_scatter <- ms_scatterchart(data = iris, x = "Sepal.Length",
  y = "Sepal.Width",  group = "Species")
my_scatter <- chart_data_fill(my_scatter,
  values = c(virginica = "#6FA2FF", versicolor = "#FF6161", setosa = "#81FF5B") )
```

---

chart_data_labels *Modify data labels settings*

Description

Data labels show details about data series. S3 generic; the default method is documented below. ChartEx charts honor a leaner set of options.

Usage

```
chart_data_labels(x, ...)

## Default S3 method:
chart_data_labels(
  x,
```

```
  num_fmt = "General",
  position = "ctr",
  show_legend_key = FALSE,
  show_val = FALSE,
  show_cat_name = FALSE,
  show_serie_name = FALSE,
  show_percent = FALSE,
  separator = ", ",
  ...
)
```

Arguments

x	an ms_chart object.
...	arguments passed to S3 methods.
num_fmt	character(1): number formatting specifies number format properties which indicate how to format and render the numeric values. It can be "General", "0.00", "#,##0", "#,##0.00", "mm-dd-yy", "m/d/yy h:mm", etc.
position	character(1): it specifies the position of the data label. It should be one of 'b', 'ctr', 'inBase', 'inEnd', 'l', 'outEnd', 'r', 't'. When grouping is 'clustered', it should be one of 'ctr','inBase','inEnd','outEnd'. When grouping is 'stacked', it should be one of 'ctr','inBase','inEnd'. When grouping is 'standard', it should be one of 'b','ctr','l','r','t'.
show_legend_key	show legend key if TRUE.
show_val	show values if TRUE.
show_cat_name	show categories if TRUE.
show_serie_name	show names of series if TRUE.
show_percent	show percentages if TRUE.
separator	separator between the label components (value, category name, series name, etc.) when multiple components are displayed. Default is ", ".

Value

An ms_chart object.

See Also

chart_labels_text(), chart_labels()

Examples

```
my_bc <- ms_barchart(
  data = browser_data, x = "browser",
  y = "value", group = "serie"
)
my_bc <- chart_data_labels(my_bc, show_val = TRUE, position = "outEnd")
```

`chart_data_line_style` *Modify line style*

Description

Specify mappings from levels in the data to displayed line style.

Usage

```
chart_data_line_style(x, values)
```

Arguments

x	an `ms_chart` object.
values	`character(num of series)`: a set of line style values to map data values to. It is a named vector, the values will be matched based on the names. Possible values are: 'none', 'solid', 'dashed', 'dotted'. If it contains only one line style, this style will be associated to all existing series.

Value

An `ms_chart` object.

See Also

Other Series customization functions: `chart_data_fill()`, `chart_data_line_width()`, `chart_data_size()`, `chart_data_smooth()`, `chart_data_stroke()`, `chart_data_symbol()`, `chart_labels_text()`

Examples

```
my_scatter <- ms_scatterchart(data = iris, x = "Sepal.Length",
  y = "Sepal.Width",  group = "Species")
my_scatter <- chart_data_fill(my_scatter,
  values = c(virginica = "#6FA2FF", versicolor = "#FF6161", setosa = "#81FF5B") )
my_scatter <- chart_data_stroke(my_scatter,
  values = c(virginica = "black", versicolor = "black", setosa = "black") )
my_scatter <- chart_data_symbol(my_scatter,
  values = c(virginica = "circle", versicolor = "diamond", setosa = "circle") )
my_scatter <- chart_data_line_style(my_scatter,
  values = c(virginica = "solid", versicolor = "dotted", setosa = "dashed") )
```

`chart_data_line_width` *Modify line width*

Description

Specify mappings from levels in the data to the displayed line width.

Usage

```
chart_data_line_width(x, values)
```

Arguments

`x`	an `ms_chart` object.
`values`	`double(num of series)`: a set of size values to map data values to. It is a named vector, the values will be matched based on the names. If it contains only one size, this size will be associated to all existing series.

Value

An `ms_chart` object.

See Also

Other Series customization functions: `chart_data_fill()`, `chart_data_line_style()`, `chart_data_size()`, `chart_data_smooth()`, `chart_data_stroke()`, `chart_data_symbol()`, `chart_labels_text()`

Examples

```
my_scatter <- ms_scatterchart(data = iris, x = "Sepal.Length",
  y = "Sepal.Width",  group = "Species")
my_scatter <- chart_settings(my_scatter, style = "lineMarker")
my_scatter <- chart_data_fill(my_scatter,
  values = c(virginica = "#6FA2FF", versicolor = "#FF6161", setosa = "#81FF5B") )
my_scatter <- chart_data_stroke(my_scatter,
  values = c(virginica = "black", versicolor = "black", setosa = "black") )
my_scatter <- chart_data_symbol(my_scatter,
  values = c(virginica = "circle", versicolor = "diamond", setosa = "circle") )
my_scatter <- chart_data_size(my_scatter,
  values = c(virginica = 20, versicolor = 16, setosa = 20) )
my_scatter <- chart_data_line_width(my_scatter,
  values = c(virginica = 2, versicolor = 3, setosa = 6) )
```

chart_data_size *Modify symbol size*

Description

Specify mappings from levels in the data to displayed size of symbols.

Usage

```
chart_data_size(x, values)
```

Arguments

x	an ms_chart object.
values	double(num of series): a set of size values to map data values to. It is a named vector, the values will be matched based on the names. If it contains only one size, this size will be associated to all existing series.

Value

An ms_chart object.

See Also

Other Series customization functions: chart_data_fill(), chart_data_line_style(), chart_data_line_width(), chart_data_smooth(), chart_data_stroke(), chart_data_symbol(), chart_labels_text()

Examples

```
my_scatter <- ms_scatterchart(data = iris, x = "Sepal.Length",
  y = "Sepal.Width",  group = "Species")
my_scatter <- chart_data_fill(my_scatter,
  values = c(virginica = "#6FA2FF", versicolor = "#FF6161", setosa = "#81FF5B") )
my_scatter <- chart_data_stroke(my_scatter,
  values = c(virginica = "black", versicolor = "black", setosa = "black") )
my_scatter <- chart_data_symbol(my_scatter,
  values = c(virginica = "circle", versicolor = "diamond", setosa = "circle") )
my_scatter <- chart_data_size(my_scatter,
  values = c(virginica = 20, versicolor = 16, setosa = 20) )
```

chart_data_smooth *Smooth series*

Description

Specify whether lines should be smoothed, per series. This feature only applies to `ms_linechart()`.

Usage

```
chart_data_smooth(x, values)
```

Arguments

x	an `ms_chart` object.
values	`integer(num of series)`: a set of smooth values to map data values to. It is a named vector, the values will be matched based on the names. Use `0` to disable smoothing and `1` to enable it. If it contains only one integer it will be associated to all existing series.

Value

An `ms_chart` object.

See Also

Other Series customization functions: `chart_data_fill()`, `chart_data_line_style()`, `chart_data_line_width()`, `chart_data_size()`, `chart_data_stroke()`, `chart_data_symbol()`, `chart_labels_text()`

Examples

```
linec <- ms_linechart(data = iris, x = "Sepal.Length",
  y = "Sepal.Width", group = "Species")
linec <- chart_data_smooth(linec,
  values = c(virginica = 0, versicolor = 0, setosa = 0) )
```

chart_data_stroke *Modify marker stroke colour*

Description

Specify mappings from levels in the data to displayed marker stroke colours.

Usage

```
chart_data_stroke(x, values, ...)
```

Arguments

x	an ms_chart object.
values	character(num of series): a set of colour values to map data values to. It is a named vector, the values will be matched based on the names. If it contains only one colour, this colour will be associated to all existing series.
...	arguments passed to S3 methods.

Value

An ms_chart object.

See Also

Other Series customization functions: chart_data_fill(), chart_data_line_style(), chart_data_line_width(), chart_data_size(), chart_data_smooth(), chart_data_symbol(), chart_labels_text()

Examples

```
my_scatter <- ms_scatterchart(data = iris, x = "Sepal.Length",
  y = "Sepal.Width",  group = "Species")
my_scatter <- chart_data_fill(my_scatter,
  values = c(virginica = "#6FA2FF", versicolor = "#FF6161", setosa = "#81FF5B") )
my_scatter <- chart_data_stroke(my_scatter,
  values = c(virginica = "black", versicolor = "black", setosa = "black") )
```

---

chart_data_symbol *Modify symbol*

Description

Specify mappings from levels in the data to displayed symbols.

Usage

```
chart_data_symbol(x, values)
```

Arguments

x	an ms_chart object.
values	character(num of series): a set of symbol values to map data values to. It is a named vector, the values will be matched based on the names. Possible values are: 'circle', 'dash', 'diamond', 'dot', 'none', 'plus', 'square', 'star', 'triangle', 'x', 'auto'. If it contains only one symbol, this symbol will be associated to all existing series.

Value

An ms_chart object.

See Also

Other Series customization functions: chart_data_fill(), chart_data_line_style(), chart_data_line_width(), chart_data_size(), chart_data_smooth(), chart_data_stroke(), chart_labels_text()

Examples

```
my_scatter <- ms_scatterchart(data = iris, x = "Sepal.Length",
  y = "Sepal.Width",  group = "Species")
my_scatter <- chart_data_fill(my_scatter,
  values = c(virginica = "#6FA2FF", versicolor = "#FF6161", setosa = "#81FF5B") )
my_scatter <- chart_data_stroke(my_scatter,
  values = c(virginica = "black", versicolor = "black", setosa = "black") )
my_scatter <- chart_data_symbol(my_scatter,
  values = c(virginica = "circle", versicolor = "diamond", setosa = "circle") )
```

---

chart_fill_ggplot2 *Apply ggplot2 color scale*

Description

The default hue color scale from ggplot2.

Usage

```
chart_fill_ggplot2(x, stroke = TRUE)
```

Arguments

x	a mschart object
stroke	a boolean. Apply the color scale to stroke? Defaults to TRUE.

Value

a mschart object

chart_fill_ggplot2()

Examples

```
p <- ms_scatterchart(
  data = iris, x = "Sepal.Length",
  y = "Sepal.Width", group = "Species"
)

p <- theme_ggplot2(p)
p <- chart_fill_ggplot2(p)
```

chart_labels *Modify axis and plot labels*

Description

Add labels to a chart, labels can be specified for x axis, y axis and plot.

Usage

```
chart_labels(x, title = NULL, xlab = NULL, ylab = NULL)
```

Arguments

x	an ms_chart object.
title	title of the chart (displayed above the plot area). Use NULL to remove it.
xlab	label for the x axis. Use NULL to remove it.
ylab	label for the y axis. Use NULL to remove it.

Value

An ms_chart object.

See Also

chart_data_labels(), chart_ax_x(), chart_ax_y()

Examples

```
mylc <- ms_linechart(
  data = browser_ts, x = "date", y = "freq",
  group = "browser"
)
mylc <- chart_labels(mylc,
  title = "my title", xlab = "my x label",
  ylab = "my y label"
)
```

chart_labels_text *Modify labels font settings*

Description

Specify mappings from levels in the data to displayed text font settings.

Usage

```
chart_labels_text(x, values)
```

Arguments

x	an ms_chart object.
values	a named list of officer::fp_text() objects to map data labels to. It is a named list, the values will be matched based on the names. If it contains only one officer::fp_text() object, it will be associated to all existing series.

Value

An ms_chart object.

See Also

Other Series customization functions: chart_data_fill(), chart_data_line_style(), chart_data_line_width(), chart_data_size(), chart_data_smooth(), chart_data_stroke(), chart_data_symbol()

Examples

```
library(officer)

fp_text_settings <- list(
  serie1 = fp_text(font.size = 7, color = "red"),
  serie2 = fp_text(font.size = 0, color = "purple"),
  serie3 = fp_text(font.size = 19, color = "wheat")
)

barchart <- ms_barchart(
  data = browser_data,
  x = "browser", y = "value", group = "serie")
barchart <- chart_data_labels(barchart, show_val = TRUE)
barchart <- chart_labels_text( barchart,
  values = fp_text_settings )
```

chart_settings *Set chart options*

Description

Set chart properties.

Usage

```
chart_settings(x, ...)

## S3 method for class 'ms_chart_ex'
chart_settings(x, ...)

## S3 method for class 'ms_barchart'
chart_settings(x, vary_colors, gap_width, dir, grouping, overlap, table, ...)

## S3 method for class 'ms_linechart'
chart_settings(x, vary_colors, style, grouping, table, ...)

## S3 method for class 'ms_areachart'
chart_settings(x, vary_colors, grouping, table, ...)

## S3 method for class 'ms_scatterchart'
chart_settings(x, vary_colors, style, ...)

## S3 method for class 'ms_stockchart'
chart_settings(
  x,
  vary_colors,
  table,
  hi_low_lines,
  up_bars_fill,
  up_bars_border,
  down_bars_fill,
  down_bars_border,
  ...
)

## S3 method for class 'ms_radarchart'
chart_settings(x, vary_colors, style, ...)

## S3 method for class 'ms_bubblechart'
chart_settings(x, vary_colors, bubble3D = FALSE, ...)

## S3 method for class 'ms_piechart'
chart_settings(x, vary_colors, hole_size, ...)
```

```
## S3 method for class 'ms_paretochart'
chart_settings(x, line, ...)

## S3 method for class 'ms_boxplotchart'
chart_settings(x, line, ...)
```

Arguments

x	an ms_chart object.
...	unused parameter
vary_colors	if TRUE, each data point in a single series is displayed in a different color.
gap_width	A gap appears between the bar or clustered bars for each category on a bar chart. The default width for this gap is 150 percent of the bar width. It can be set between 0 and 500 percent of the bar width.
dir	the direction of the bars in the chart, value must be one of "horizontal" or "vertical".
grouping	grouping of the series. For a barchart one of "percentStacked", "clustered", "standard" or "stacked". For a linechart or an areachart one of "percentStacked", "standard" or "stacked" ("clustered" is bar-only).
overlap	In a bar chart having two or more series, the bars for each category are clustered together. By default, these bars are directly adjacent to each other. The bars can be made to overlap each other or have a space between them using the overlap property. Its values range between -100 and 100, representing the percentage of the bar width by which to overlap adjacent bars. A setting of -100 creates a gap of a full bar width and a setting of 100 causes all the bars in a category to be superimposed. The default value is 0.
table	if TRUE set a table below the barchart.
style	Style for the linechart or scatterchart type of markers. One of 'none', 'line', 'lineMarker', 'marker', 'smooth', 'smoothMarker'.
hi_low_lines	an officer::fp_border() for the high-low lines. Set to FALSE to hide them.
up_bars_fill	fill colour for up bars (OHLC only, close > open).
up_bars_border	an officer::fp_border() for up bar borders.
down_bars_fill	fill colour for down bars (OHLC only, close < open).
down_bars_border	an officer::fp_border() for down bar borders.
bubble3D	logical, use 3D effect for bubbles.
hole_size	size of the hole in a doughnut chart, between 0 and 90 (percent of the radius). Default 0 produces a pie chart; values above 0 produce a doughnut chart.
line	stroke for the cumulative percentage line. One of: NULL (default, matches Excel-native: theme accent2 colour scaled to chart palette), FALSE to suppress the line override (line then depends on the chartstyle sidecar and may render as invisible), or an officer::fp_border().

Value

An ms_chart object.

Methods (by class)

- chart_settings(ms_chart_ex): fallback for chartEx types that expose no settings (funnel, histogram, sunburst, treemap, waterfall). Replaces the default no applicable method error with a discoverable message.
- chart_settings(ms_barchart): barchart settings
- chart_settings(ms_linechart): linechart settings
- chart_settings(ms_areachart): areachart settings
- chart_settings(ms_scatterchart): scatterchart settings
- chart_settings(ms_stockchart): stockchart settings
- chart_settings(ms_radarchart): radarchart settings
- chart_settings(ms_bubblechart): bubblechart settings
- chart_settings(ms_piechart): piechart settings
- chart_settings(ms_paretochart): paretochart settings
- chart_settings(ms_boxplotchart): boxplotchart settings

Illustrations

See Also

ms_barchart(), ms_areachart(), ms_scatterchart(), ms_linechart()

Examples

```
library(mschart)
library(officer)

chart_01 <- ms_barchart(
  data = browser_data, x = "browser",
  y = "value", group = "serie"
)
chart_01 <- chart_theme(chart_01,
  grid_major_line_x = fp_border(width = 0),
  grid_minor_line_x = fp_border(width = 0)
)

chart_02 <- chart_settings(
  x = chart_01,
  grouping = "stacked", overlap = 100
)
```

```
chart_03 <- ms_areachart(
  data = browser_ts, x = "date",
  y = "freq", group = "browser"
)
chart_03 <- chart_settings(chart_03,
  grouping = "percentStacked"
)
```

`chart_table` *Data table settings*

Description

Define visual settings for the data table displayed below the chart. Requires `chart_settings(x, table = TRUE)` to be called first.

Usage

```
chart_table(x, horizontal, vertical, outline, show_keys)
```

Arguments

`x`	an `ms_chart` object.
`horizontal`	write horizontal lines in the table
`vertical`	write vertical lines in the table
`outline`	write an outline in the table
`show_keys`	show keys in the table

Value

An `ms_chart` object.

See Also

`chart_settings()`

Examples

```
data <- data.frame(
  supp = factor(rep(c("OJ", "VC"), each = 3),
                levels = c("OJ", "VC")),
  dose = factor(rep(c("low", "medium", "high"), 2),
                levels = c("low", "medium", "high")),
  length = c(13.23, 22.7, 24.06, 7.98, 16.77, 26.14),
  label = LETTERS[1:6],
  stringsAsFactors = FALSE
)
```

```
# example chart 03 -------
chart <- ms_linechart(
  data = data, x = "dose", y = "length",
  group = "supp", labels = "label"
)
chart <- chart_settings(
  x = chart, table = TRUE
)

chart <- chart_table(chart,
  horizontal = TRUE, vertical = FALSE,
  outline = TRUE, show_keys = FALSE
)
```

`mschart` *Chart Generation for 'Microsoft Word', 'Microsoft Excel' and 'Microsoft PowerPoint' Documents*

Description

It lets R users create Microsoft Office charts from data, and then add title, legends, and annotations to the chart object.

The graph produced is a Microsoft graph, which means that it can be edited in your Microsoft software and that the underlying data are available.

The package will not allow you to make the same charts as with ggplot2. It allows only a subset of the charts possible with 'Office Chart'. The package is often used to industrialize graphs that are then consumed and annotated by non-R users.

The following chart types are available.

Classical charts (Office 2007+ DrawingML pipeline):

- bar charts: `ms_barchart()`
- line charts: `ms_linechart()`
- scatter plots: `ms_scatterchart()`
- area charts: `ms_areachart()`
- pie and doughnut charts: `ms_piechart()`
- bubble charts: `ms_bubblechart()`
- radar (spider) charts: `ms_radarchart()`
- stock charts (HLC and OHLC): `ms_stockchart()`

chartEx charts (Office 2016+ pipeline; older viewers show a placeholder):

- box-and-whisker: `ms_boxplotchart()`
- funnel: `ms_funnelchart()`
- histogram: `ms_histogramchart()`

- pareto: ms_paretochart()
- sunburst: ms_sunburstchart()
- treemap: ms_treemapchart()
- waterfall: ms_waterfallchart()

Several classical chart types can be combined on a single chart, with an optional secondary axis (y or x), using ms_chart_combine().

These functions create a 'chart' object that can be customized:

- by using options specific to the chart (with chart_settings()),
- by changing the options related to the axes (with chart_ax_x() and chart_ax_y()),
- by changing the options related to the labels (with chart_data_labels()),
- by changing the colors, line widths, ... with functions
  - chart_labels_text()
  - chart_data_fill()
  - chart_data_line_style()
  - chart_data_line_width()
  - chart_data_size()
  - chart_data_smooth()
  - chart_data_stroke()
  - chart_data_symbol()
- by changing the general theme with function chart_theme(),
- by changing the title labels with function chart_labels().

You can add a chart into a slide in PowerPoint with function ph_with.ms_chart().

You can add a chart into a Word document with function body_add_chart().

You can add a chart into an Excel sheet with function sheet_add_drawing.ms_chart().

Series styling properties by chart type:

Not all series styling properties have an effect on every chart type. A warning is emitted when a property is set on a chart type that does not support it.

Classical charts:

Property	bar	line	area	scatter	stock	radar	bubble	pie
fill	x	x	x	x	x	x	x	x
colour	x	x	x	x	x	x	x	x
symbol		x		x	x	x		
size		x		x	x	x		
line_width	x	x	x	x	x	x	x	x
line_style		x		x	x	x		
smooth		x		x				
labels_fp	x	x	x	x		x		x

chartEx charts expose a narrower set of styling knobs, since most visual aspects are computed by Office from the data (bins, levels, connectors, ...). For these, prefer the chart-type-specific

`chart_settings()` method when one is available (e.g. for pareto and boxplot: `chart_settings(x, line = fp_border(...))`).

Property	boxplot	funnel	histogram	pareto	sunburst	treemap	waterfall
fill	x	x	x	x	x	x	x
colour	x	x	x	x	x	x	x
labels_fp	x	x	x	x	x	x	x

Two arguments often confused: `asis` and `write_data`:

`asis` is a **constructor argument** (on `ms_barchart()`, `ms_linechart()` and most other classical constructors). It describes the *input shape* of the data frame:

- `asis = FALSE` (default): long format, with a `group` column that splits the rows into series. `mschart` reshapes the data internally.
- `asis = TRUE`: wide format, with one column per series. `y` accepts a vector of series column names.

`write_data` is an **embed-time argument** of `sheet_add_drawing.ms_chart()` for the Excel pipeline. It decides whether `mschart` writes the chart's data into the target sheet (`TRUE`, the default) or leaves you in charge of placing it via `officer::sheet_write_data()` (`FALSE`, recommended for non-trivial workbooks).

The two arguments are independent: a chart built with `asis = TRUE` can still be embedded with either `write_data = TRUE` or `write_data = FALSE`.

Author(s)

Maintainer: David Gohel <david.gohel@ardata.fr>

Authors:

- David Gohel <david.gohel@ardata.fr>

Other contributors:

- ArData [copyright holder]
- YouGov [funder]
- Jan Marvin Garbuszus (support for openxlsx2 and combo charts) [contributor]
- Stefan Moog <moogs@gmx.de> (support to set chart and plot area color and border, fix for line style none) [contributor]
- Eli Daniels <eli.daniels@ardata.fr> [contributor]
- Marlon Molina (added table feature) [contributor]
- Rokas Klydzia (custom labels) [contributor]
- David Camposeco <david.camposeco.paulsen@gmail.com> (chart_data_smooth function) [contributor]
- Dan Joplin (fix scatter plot data structure) [contributor]

See Also

https://ardata-fr.github.io/officeverse/

`ms_areachart` *Areachart object*

Description

Creation of an areachart object that can be inserted in a 'Microsoft' document.

Area charts can be used to plot change over time and draw attention to the total value across a trend. By showing the sum of the plotted values, an area chart also shows the relationship of parts to a whole.

Usage

```
ms_areachart(data, x, y, group = NULL, labels = NULL, asis = FALSE)
```

Arguments

`data`	a data.frame
`x`	column name for x values.
`y`	column name for y values.
`group`	grouping column name used to split data into series. Optional.
`labels`	column names of columns to be used as custom data labels displayed next to data points (not axis labels). Optional. If more than one name is provided, only the first one will be used as a label, but all labels (transposed if a group is used) will be available in the Excel file associated with the chart.
`asis`	logical parameter defaulting to FALSE. When FALSE, the data is reshaped internally so that each series becomes a separate column. When TRUE, the data is used as-is and must already have one column for categories and one column per series, and `y` accepts a vector of series column names. `asis` describes the *input shape* read by the constructor. Not to be confused with the `write_data` argument of `sheet_add_drawing.ms_chart()`, which controls whether `mschart` writes the chart's data into an Excel sheet at embed time. The two are independent.

Value

An `ms_chart` object.

See Also

`chart_settings()`, `chart_ax_x()`, `chart_ax_y()`, `chart_data_labels()`, `chart_theme()`, `chart_labels()`

Other 'Office' chart objects: `ms_barchart()`, `ms_boxplotchart()`, `ms_bubblechart()`, `ms_chart_combine()`, `ms_funnelchart()`, `ms_histogramchart()`, `ms_linechart()`, `ms_paretochart()`, `ms_piechart()`, `ms_radarchart()`, `ms_scatterchart()`, `ms_stockchart()`, `ms_sunburstchart()`, `ms_treemapchart()`, `ms_waterfallchart()`

Examples

```
library(officer)
mytheme <- mschart_theme(
  axis_title_x = fp_text(color = "red", font.size = 24, bold = TRUE),
  axis_title_y = fp_text(color = "green", font.size = 12, italic = TRUE),
  grid_major_line_y = fp_border(width = 1, color = "orange"),
  axis_ticks_y = fp_border(width = 1, color = "orange")
)

# example ac_01 -------
ac_01 <- ms_areachart(
  data = browser_ts, x = "date",
  y = "freq", group = "browser"
)
ac_01 <- chart_ax_y(ac_01, cross_between = "between", num_fmt = "General")
ac_01 <- chart_ax_x(ac_01, cross_between = "midCat", num_fmt = "m/d/yy")
ac_01 <- set_theme(ac_01, mytheme)


# example ac_02 -------
ac_02 <- chart_settings(ac_01, grouping = "percentStacked")

# example ac_03 -------
ac_03 <- chart_settings(ac_01, grouping = "percentStacked", table = TRUE)
ac_03 <- chart_table(
  ac_03,
  horizontal = FALSE, vertical = FALSE,
  outline = FALSE, show_keys = TRUE)
```

---

ms_barchart *Barchart object*

Description

Creation of a barchart object that can be inserted in a 'Microsoft' document.

Bar charts illustrate comparisons among individual items. In a bar chart, the categories are typically organized along the vertical axis, and the values along the horizontal axis.

Consider using a bar chart when:

- The axis labels are long.
- The values that are shown are durations.

Usage

```
ms_barchart(data, x, y, group = NULL, labels = NULL, asis = FALSE)
```

Arguments

data	a data.frame
x	column name for x values.
y	column name for y values.
group	grouping column name used to split data into series. Optional.
labels	column names of columns to be used as custom data labels displayed next to data points (not axis labels). Optional. If more than one name is provided, only the first one will be used as a label, but all labels (transposed if a group is used) will be available in the Excel file associated with the chart.
asis	logical parameter defaulting to FALSE. When FALSE, the data is reshaped internally so that each series becomes a separate column. When TRUE, the data is used as-is and must already have one column for categories and one column per series, and y accepts a vector of series column names. asis describes the *input shape* read by the constructor. Not to be confused with the write_data argument of sheet_add_drawing.ms_chart(), which controls whether mschart writes the chart's data into an Excel sheet at embed time. The two are independent.

Value

An ms_chart object.

Illustrations

See Also

chart_settings(), chart_ax_x(), chart_ax_y(), chart_data_labels(), chart_theme(), chart_labels()

Other 'Office' chart objects: ms_areachart(), ms_boxplotchart(), ms_bubblechart(), ms_chart_combine(), ms_funnelchart(), ms_histogramchart(), ms_linechart(), ms_paretochart(), ms_piechart(), ms_radarchart(), ms_scatterchart(), ms_stockchart(), ms_sunburstchart(), ms_treemapchart(), ms_waterfallchart()

Examples

```
library(officer)
library(mschart)
library(officer)


# example chart 01 -------

chart_01 <- ms_barchart(
  data = browser_data, x = "browser",
  y = "value", group = "serie"
)
chart_01 <- chart_settings(
```

```
  x = chart_01, dir = "vertical",
  grouping = "clustered", gap_width = 50
)
chart_01 <- chart_ax_x(
  x = chart_01, cross_between = "between",
  major_tick_mark = "out"
)
chart_01 <- chart_ax_y(
  x = chart_01, cross_between = "midCat",
  major_tick_mark = "in"
)
# print(chart_01, preview = TRUE)


# example chart 02 -------
dat <- data.frame(
  Species = factor(c("setosa", "versicolor", "virginica"),
    levels = c("setosa", "versicolor", "virginica")
  ),
  mean = c(5.006, 5.936, 6.588)
)
chart_02 <- ms_barchart(data = dat, x = "Species", y = "mean")
chart_02 <- chart_settings(x = chart_02, dir = "horizontal")
chart_02 <- chart_theme(x = chart_02, title_x_rot = 270, title_y_rot = 0)
# print(chart_02, preview = TRUE)



# example chart 03 -------

mytheme <- mschart_theme(
  axis_title_x = fp_text(color = "gray", font.size = 20, bold = TRUE),
  axis_title_y = fp_text(color = "gray", font.size = 20, italic = TRUE),
  grid_major_line_y = fp_border(width = 1, color = "wheat"),
  axis_ticks_y = fp_border(width = 1, color = "gray")
)

chart_03 <- ms_barchart(
  data = browser_data, x = "browser",
  y = "value", group = "serie"
)
chart_03 <- chart_settings(chart_03,
  grouping = "stacked",
  gap_width = 150, overlap = 100
)
chart_03 <- chart_ax_x(chart_03,
  cross_between = "between",
  major_tick_mark = "out", minor_tick_mark = "none"
)
chart_03 <- chart_ax_y(chart_03,
  num_fmt = "0.00",
  minor_tick_mark = "none"
)
```

```
chart_03 <- set_theme(chart_03, mytheme)
chart_03 <- chart_labels(x = chart_03, title = "Things in percent")
chart_03 <- chart_data_labels(chart_03,
  position = "ctr",
  show_val = TRUE
)
chart_03 <- chart_labels_text(chart_03,
  fp_text(color = "white", bold = TRUE, font.size = 9)
)
chart_03 <- chart_data_fill(chart_03,
  values = c(
    serie1 = "#4477AA",
    serie2 = "#CC6677",
    serie3 = "#DDCC77"
  )
)
chart_03 <- chart_data_stroke(chart_03,
  values = c(
    serie1 = "#223B55",
    serie2 = "#66333C",
    serie3 = "#6F663C"
  )
)
chart_03 <- chart_data_line_width(chart_03,
  values = c(serie1 = 2, serie2 = 2, serie3 = 2)
)
# print(chart_03, preview = TRUE)

# example chart 04 -------

dat_groups <-
  data.frame(
    cut = c(
      "Fair", "Fair", "Fair", "Fair", "Fair",
      "Fair", "Fair", "Fair", "Good", "Good", "Good", "Good", "Good",
      "Good", "Good", "Good", "Very Good", "Very Good", "Very Good",
      "Very Good", "Very Good", "Very Good", "Very Good", "Very Good",
      "Premium", "Premium", "Premium", "Premium", "Premium",
      "Premium", "Premium", "Premium", "Ideal", "Ideal", "Ideal", "Ideal",
      "Ideal", "Ideal", "Ideal", "Ideal"
    ),
    clarity = c(
      "I1", "SI2", "SI1", "VS2", "VS1", "VVS2",
      "VVS1", "IF", "I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1",
      "IF", "I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF",
      "I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF", "I1",
      "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF"
    ),
    carat = c(
      1.065, 1.01, 0.98, 0.9, 0.77, 0.7, 0.7,
      0.47, 1.07, 1, 0.79, 0.82, 0.7, 0.505, 0.4, 0.46, 1.145, 1.01,
      0.77, 0.71, 0.7, 0.4, 0.36, 0.495, 1.11, 1.04, 0.9, 0.72, 0.7,
      0.455, 0.4, 0.36, 1.13, 1, 0.71, 0.53, 0.53, 0.44, 0.4, 0.34
```

```
    ),
    n = c(
      210L, 466L, 408L, 261L, 170L, 69L, 17L, 9L,
      96L, 1081L, 1560L, 978L, 648L, 286L, 186L, 71L, 84L, 2100L,
      3240L, 2591L, 1775L, 1235L, 789L, 268L, 205L, 2949L, 3575L, 3357L,
      1989L, 870L, 616L, 230L, 146L, 2598L, 4282L, 5071L, 3589L,
      2606L, 2047L, 1212L
    )
  )

dat_groups$label <- sprintf(
  "carat median is %.01f",
  dat_groups$carat
)
dat_groups

text_prop <- fp_text(font.size = 11, color = "gray")

chart_04 <- ms_barchart(
  data = dat_groups, x = "cut",
  labels = "label", y = "n", group = "clarity"
)
chart_04 <- chart_settings(chart_04,
  grouping = "clustered", dir = "horizontal",
  gap_width = 0
)
chart_04 <- chart_data_labels(chart_04, position = "outEnd")
chart_04 <- chart_labels_text(chart_04, text_prop)
chart_04 <- chart_theme(chart_04, title_x_rot = 270, title_y_rot = 0)
# print(chart_04, preview = TRUE)

# example chart 05 -------

dat_no_group <- data.frame(
  stringsAsFactors = FALSE,
  cut = c("Fair", "Good", "Very Good", "Premium", "Ideal"),
  carat = c(1, 0.82, 0.71, 0.86, 0.54),
  n = c(1610L, 4906L, 12082L, 13791L, 21551L),
  label = c(
    "carat median is 1.0",
    "carat median is 0.8", "carat median is 0.7",
    "carat median is 0.9", "carat median is 0.5"
  )
)
chart_05 <- ms_barchart(
  data = dat_no_group,
  x = "cut", labels = "label", y = "n"
)
chart_05 <- chart_settings(chart_05,
  grouping = "clustered"
)
chart_05 <- chart_data_labels(chart_05, position = "outEnd")
chart_05 <- chart_labels_text(chart_05, text_prop)
```

```
# print(chart_05, preview = TRUE)

# example chart 06 -------
chart_06 <- ms_barchart(
  data = dat_no_group,
  x = "cut", labels = "label", y = "n"
)
chart_06 <- chart_settings(chart_06,
  grouping = "clustered", table = TRUE
)
chart_06 <- chart_data_labels(chart_06, position = "outEnd")
chart_06 <- chart_labels_text(chart_06, text_prop)
# print(chart_06, preview = TRUE)


# example chart 07 -------
# Series order (and therefore legend order) follows the levels of
# the `group` factor. Convert the column to a factor with the desired
# level order before passing it to ms_barchart().
ordered_data <- browser_data
ordered_data$serie <- factor(ordered_data$serie,
  levels = c("serie3", "serie1", "serie2")
)
chart_07 <- ms_barchart(
  data = ordered_data, x = "browser",
  y = "value", group = "serie"
)
chart_07 <- chart_labels(chart_07,
  title = "Series and legend ordered via factor levels"
)
# print(chart_07, preview = TRUE)

# example chart 08 -------
# Wide-format input: each series is its own column. Pass the series
# column names as a vector in `y` and set `asis = TRUE`.
browser_wide <- data.frame(
  browser = unique(browser_data$browser),
  serie1  = browser_data$value[browser_data$serie == "serie1"],
  serie2  = browser_data$value[browser_data$serie == "serie2"],
  serie3  = browser_data$value[browser_data$serie == "serie3"]
)
chart_08 <- ms_barchart(
  data = browser_wide,
  x = "browser",
  y = c("serie1", "serie2", "serie3"),
  asis = TRUE
)
# print(chart_08, preview = TRUE)
```

---

ms_boxplotchart *Boxplot chart object*

Description

Creation of a box-and-whisker chart object that can be inserted in a 'Microsoft' document. Boxplot charts use the chartEx pipeline (Office 2016+); older versions of 'Microsoft Office' will display a fallback placeholder.

Data is in long format: one row per observation. Office computes quartiles and whiskers from the raw values; do not pre-aggregate.

Usage

```
ms_boxplotchart(
  data,
  x,
  y,
  quartile_method = c("exclusive", "inclusive"),
  show_mean_marker = TRUE,
  show_mean_line = FALSE,
  show_outliers = TRUE,
  show_inner_points = FALSE
)
```

Arguments

`data`	a data.frame.
`x`	category column name. Each unique value becomes one box.
`y`	numeric value column name (raw observations).
`quartile_method`	one of `"exclusive"` (default) or `"inclusive"`. Affects how Q1/Q3 are computed when the count is even.
`show_mean_marker`	logical, draw the mean as a marker. Default TRUE.
`show_mean_line`	logical, draw a line connecting means across boxes. Default FALSE.
`show_outliers`	logical, plot outlier points. Default TRUE.
`show_inner_points`	logical, plot all non-outlier points. Default FALSE.

Value

An `ms_chart` object (subclass `ms_boxplotchart`).

See Also

`chart_labels()`

Other 'Office' chart objects: `ms_areachart()`, `ms_barchart()`, `ms_bubblechart()`, `ms_chart_combine()`, `ms_funnelchart()`, `ms_histogramchart()`, `ms_linechart()`, `ms_paretochart()`, `ms_piechart()`, `ms_radarchart()`, `ms_scatterchart()`, `ms_stockchart()`, `ms_sunburstchart()`, `ms_treemapchart()`, `ms_waterfallchart()`

Examples

```
library(officer)

set.seed(1)
dat <- data.frame(
  group = rep(c("A", "B", "C"), each = 20),
  value = c(rnorm(20, 0, 5), rnorm(20, 3, 7), rnorm(20, -2, 4))
)
bp <- ms_boxplotchart(dat, x = "group", y = "value")

doc <- read_pptx()
doc <- add_slide(doc)
doc <- ph_with(doc, bp, location = ph_location_fullsize())
print(doc, target = tempfile(fileext = ".pptx"))
```

---

ms_bubblechart *Bubblechart object*

Description

Creation of a bubblechart object that can be inserted in a 'Microsoft' document. A bubble chart is a scatter chart where each point has a third numeric dimension controlling its size.

Usage

```
ms_bubblechart(data, x, y, size, group = NULL, labels = NULL, asis = FALSE)
```

Arguments

data	a data.frame
x	column name for x values.
y	column name for y values.
size	column name for bubble size values (must be numeric)
group	grouping column name used to split data into series. Optional.
labels	column names of columns to be used as custom data labels displayed next to data points (not axis labels). Optional. If more than one name is provided, only the first one will be used as a label, but all labels (transposed if a group is used) will be available in the Excel file associated with the chart.
asis	logical parameter defaulting to FALSE. When FALSE, the data is reshaped internally so that each series becomes a separate column. When TRUE, the data is used as-is and must already have one column for categories and one column per series, and y accepts a vector of series column names. asis describes the *input shape* read by the constructor. Not to be confused with the write_data argument of sheet_add_drawing.ms_chart(), which controls whether mschart writes the chart's data into an Excel sheet at embed time. The two are independent.

Value

An ms_chart object.

See Also

chart_settings(), chart_ax_x(), chart_ax_y(), chart_data_labels(), chart_theme(), chart_labels()

Other 'Office' chart objects: ms_areachart(), ms_barchart(), ms_boxplotchart(), ms_chart_combine(), ms_funnelchart(), ms_histogramchart(), ms_linechart(), ms_paretochart(), ms_piechart(), ms_radarchart(), ms_scatterchart(), ms_stockchart(), ms_sunburstchart(), ms_treemapchart(), ms_waterfallchart()

Examples

```
library(officer)

dat <- data.frame(
  x = c(1, 2, 3, 4, 5),
  y = c(10, 20, 15, 25, 30),
  sz = c(5, 10, 7, 15, 12),
  grp = rep("s1", 5)
)

bubble <- ms_bubblechart(
  data = dat, x = "x", y = "y",
  size = "sz", group = "grp"
)

# adjust axes to avoid clipping extreme bubbles
bubble <- chart_ax_x(bubble, limit_min = 0, limit_max = 6)
bubble <- chart_ax_y(bubble, limit_min = 5, limit_max = 35)
bubble
```

---

ms_chart_combine *Combined chart object*

Description

Combine several chart objects into a single chart with shared axes. Each chart must be a named argument.

The title and x-axis label are taken from the first chart. The y-axis label of the first chart on the secondary axis is used as the secondary y-axis label.

Only one secondary y-axis (right) and one secondary x-axis (top) are supported.

Usage

```
ms_chart_combine(..., secondary_y = NULL, secondary_x = NULL)
```

Arguments

...	named ms_chart objects.
secondary_y	character vector of chart names to plot on the secondary (right) y-axis.
secondary_x	character vector of chart names to plot on the secondary (top) x-axis.

See Also

chart_settings(), chart_ax_x(), chart_ax_y(), chart_data_labels(), chart_theme(), chart_labels()

Other 'Office' chart objects: ms_areachart(), ms_barchart(), ms_boxplotchart(), ms_bubblechart(), ms_funnelchart(), ms_histogramchart(), ms_linechart(), ms_paretochart(), ms_piechart(), ms_radarchart(), ms_scatterchart(), ms_stockchart(), ms_sunburstchart(), ms_treemapchart(), ms_waterfallchart()

Examples

```
# example chart 01 -------

dat <- longley
dat$Year <- as.Date(paste0(dat$Year, "-01-01"))

dat_empl <- data.frame(
  Year = rep(dat$Year, 2),
  value = c(dat$Employed, dat$Unemployed),
  serie = rep(c("Employed", "Unemployed"), each = nrow(dat)),
  stringsAsFactors = FALSE
)

empl <- ms_barchart(
  data = dat_empl, x = "Year",
  y = "value", group = "serie"
)
empl <- chart_labels(
  x = empl, title = "Longley's Economic Regression Data",
  xlab = "1947 to 1962", ylab = "Employment"
)
empl <- chart_ax_x(x = empl, num_fmt = "yyyy")
empl <- as_bar_stack(x = empl, dir = "vertical")

gdp <- ms_linechart(data = dat, x = "Year", y = "GNP.deflator")
gdp <- chart_labels(x = gdp, ylab = "GNP implicit price deflator (1954 = 100)")

cb <- ms_chart_combine(empl = empl, gdp = gdp, secondary_y = "gdp")

# example chart 02 -------
# stacked bars + target line on a shared axis (no secondary axis).

browser_data <- data.frame(
  browser = rep(c("Android", "Chrome", "IE", "Firefox", "Opera", "Safari"),
                each = 3),
  serie   = rep(c("v1", "v2", "v3"), times = 6),
```

```
  value   = c(80, 90, 77, 150, 140, 131, 60, 70, 62,
              50, 60, 52, 40, 50, 42, 55, 65, 55)
)

target <- data.frame(
  browser = unique(browser_data$browser),
  cible   = 220
)

bars <- ms_barchart(browser_data, x = "browser",
                    y = "value", group = "serie")
bars <- as_bar_stack(bars)

line <- ms_linechart(target, x = "browser", y = "cible")
line <- chart_data_stroke(line, values = "#D62728")
line <- chart_data_line_width(line, values = 2)
line <- chart_data_symbol(line, values = "none")

cb2 <- ms_chart_combine(bars = bars, target = line)

# example chart 03 -------
# two scatter charts on a shared y axis with independent x ranges.
# bottom (primary) chart uses temperature in Celsius;
# top (secondary_x) chart uses pressure in hPa.
# y series must have distinct column names (no collision in the
# embedded sheet) — here humidity_t vs humidity_p.

set.seed(1)
n <- 20
weather <- data.frame(
  temperature_c = runif(n, 5, 30),
  pressure_hpa  = runif(n, 990, 1030),
  humidity_t    = runif(n, 30, 90),
  humidity_p    = runif(n, 30, 90)
)

sc_temp <- ms_scatterchart(
  data = weather, x = "temperature_c", y = "humidity_t"
)
sc_temp <- chart_labels(
  sc_temp, title = "Humidity vs temperature and pressure",
  xlab = "Temperature (°C)", ylab = "Humidity (%)"
)

sc_pres <- ms_scatterchart(
  data = weather, x = "pressure_hpa", y = "humidity_p"
)
sc_pres <- chart_data_symbol(sc_pres, values = "triangle")
sc_pres <- chart_labels(sc_pres, xlab = "Pressure (hPa)")

cb3 <- ms_chart_combine(
  temp = sc_temp, pres = sc_pres,
  secondary_x = "pres"
```

```
)
```

`ms_funnelchart` *Funnel chart object*

Description

Creation of a funnel chart object that can be inserted in a 'Microsoft' document. Funnel charts use the chartEx pipeline (Office 2016+); older versions of 'Microsoft Office' will display a fallback placeholder.

Each row is one stage of the funnel. Values are typically decreasing (e.g. visitors -> leads -> customers). Bars are centered horizontally and width is proportional to the value.

Usage

```
ms_funnelchart(data, x, y)
```

Arguments

`data`	a data.frame.
`x`	category column name (stage label).
`y`	numeric value column name.

Value

An `ms_chart` object (subclass `ms_funnelchart`).

See Also

`chart_labels()`

Other 'Office' chart objects: `ms_areachart()`, `ms_barchart()`, `ms_boxplotchart()`, `ms_bubblechart()`, `ms_chart_combine()`, `ms_histogramchart()`, `ms_linechart()`, `ms_paretochart()`, `ms_piechart()`, `ms_radarchart()`, `ms_scatterchart()`, `ms_stockchart()`, `ms_sunburstchart()`, `ms_treemapchart()`, `ms_waterfallchart()`

Examples

```
library(officer)

dat <- data.frame(
  stage = c("Visitors", "Leads", "Opportunities", "Quotes", "Customers"),
  count = c(5000, 4000, 3000, 1000, 250),
  stringsAsFactors = FALSE
)
fn <- ms_funnelchart(data = dat, x = "stage", y = "count")

doc <- read_pptx()
```

```
doc <- add_slide(doc)
doc <- ph_with(doc, fn, location = ph_location_fullsize())
print(doc, target = tempfile(fileext = ".pptx"))
```

`ms_histogramchart` *Histogram chart object*

Description

Creation of a histogram chart object that can be inserted in a 'Microsoft' document. Histogram charts use the chartEx pipeline (Office 2016+); older versions of 'Microsoft Office' will display a fallback placeholder.

Data is a single column of raw numeric observations. Office computes the bins automatically; pass `bin_count` or `bin_width` to override.

Usage

```
ms_histogramchart(
  data,
  value,
  bin_count = NULL,
  bin_width = NULL,
  interval_closed = c("right", "left"),
  underflow = NULL,
  overflow = NULL
)
```

Arguments

`data`	a data.frame.
`value`	numeric column name (raw observations).
`bin_count`	integer, requested number of bins. Mutually exclusive with `bin_width`. NULL = automatic.
`bin_width`	numeric, requested bin width. Mutually exclusive with `bin_count`. NULL = automatic.
`interval_closed`	one of `"right"` (default) or `"left"`. Defines which end of each bin is inclusive.
`underflow`	numeric, values below this go in an "underflow" bin. NULL to disable.
`overflow`	numeric, values above this go in an "overflow" bin. NULL to disable.

Value

An `ms_chart` object (subclass `ms_histogramchart`).

See Also

chart_labels()

Other 'Office' chart objects: ms_areachart(), ms_barchart(), ms_boxplotchart(), ms_bubblechart(), ms_chart_combine(), ms_funnelchart(), ms_linechart(), ms_paretochart(), ms_piechart(), ms_radarchart(), ms_scatterchart(), ms_stockchart(), ms_sunburstchart(), ms_treemapchart(), ms_waterfallchart()

Examples

```
library(officer)

set.seed(1)
dat <- data.frame(x = rnorm(200, mean = 10, sd = 4))
hi <- ms_histogramchart(dat, value = "x", bin_count = 12)

doc <- read_pptx()
doc <- add_slide(doc)
doc <- ph_with(doc, hi, location = ph_location_fullsize())
print(doc, target = tempfile(fileext = ".pptx"))
```

---

ms_linechart *Linechart object*

Description

Creation of a linechart object that can be inserted in a 'Microsoft' document.

In a line chart, category data is distributed evenly along the horizontal axis, and all value data is distributed evenly along the vertical axis. Line charts can show continuous data over time on an evenly scaled axis, so they're ideal for showing trends in data at equal intervals, like months and quarters.

Usage

```
ms_linechart(data, x, y, group = NULL, labels = NULL, asis = FALSE)
```

Arguments

data	a data.frame
x	column name for x values.
y	column name for y values.
group	grouping column name used to split data into series. Optional.
labels	column names of columns to be used as custom data labels displayed next to data points (not axis labels). Optional. If more than one name is provided, only the first one will be used as a label, but all labels (transposed if a group is used) will be available in the Excel file associated with the chart.

asis — logical parameter defaulting to FALSE. When FALSE, the data is reshaped internally so that each series becomes a separate column. When TRUE, the data is used as-is and must already have one column for categories and one column per series, and y accepts a vector of series column names. asis describes the *input shape* read by the constructor. Not to be confused with the write_data argument of sheet_add_drawing.ms_chart(), which controls whether mschart writes the chart's data into an Excel sheet at embed time. The two are independent.

Value

An ms_chart object.

Illustrations

See Also

chart_settings(), chart_ax_x(), chart_ax_y(), chart_data_labels(), chart_theme(), chart_labels()

Other 'Office' chart objects: ms_areachart(), ms_barchart(), ms_boxplotchart(), ms_bubblechart(), ms_chart_combine(), ms_funnelchart(), ms_histogramchart(), ms_paretochart(), ms_piechart(), ms_radarchart(), ms_scatterchart(), ms_stockchart(), ms_sunburstchart(), ms_treemapchart(), ms_waterfallchart()

Examples

```
library(officer)
# example chart_01 -------
chart_01 <- ms_linechart(
  data = us_indus_prod,
  x = "date", y = "value",
  group = "type"
)

chart_01 <- chart_ax_x(
  x = chart_01, num_fmt = "[$-fr-FR]mmm yyyy",
  limit_min = min(us_indus_prod$date), limit_max = as.Date("1992-01-01")
)

chart_01 <- chart_data_stroke(
  x = chart_01,
  values = c(adjusted = "red", unadjusted = "gray")
)

chart_01 <- chart_data_line_width(
  x = chart_01,
  values = c(adjusted = 2, unadjusted = 5)
)

chart_01 <- chart_theme(chart_01,
```

```
  grid_major_line_x = fp_border(width = 0),
  grid_minor_line_x = fp_border(width = 0)
)

# example chart_02 -------
data <- data.frame(
  supp = factor(rep(c("OJ", "VC"), each = 3), levels = c("OJ", "VC")),
  dose = factor(rep(c("low", "medium", "high"), 2), levels = c("low", "medium", "high")),
  length = c(13.23, 22.7, 24.06, 7.98, 16.77, 26.14),
  label = LETTERS[1:6],
  stringsAsFactors = FALSE
)

chart_02 <- ms_linechart(
  data = data, x = "dose", y = "length",
  group = "supp", labels = "label"
)
chart_02 <- chart_ax_y(
  x = chart_02, cross_between = "between",
  limit_min = 5, limit_max = 30,
  num_fmt = "General"
)
chart_02 <- chart_data_labels(
  x = chart_02, position = "l"
)

# example chart 03 -------
chart_03 <- ms_linechart(
  data = data, x = "dose", y = "length",
  group = "supp", labels = "label"
)
chart_03 <- chart_ax_y(
  x = chart_03, cross_between = "between",
  limit_min = 5, limit_max = 30,
  num_fmt = "General"
)
chart_03 <- chart_data_labels(
  x = chart_03, position = "l"
)

chart_03 <- chart_settings(
  x = chart_03, table = TRUE
)

chart_03 <- chart_table(chart_03,
  horizontal = TRUE, vertical = FALSE,
  outline = TRUE, show_keys = FALSE
)
```

`ms_paretochart` *Pareto chart object*

Description

Creation of a Pareto chart object that can be inserted in a 'Microsoft' document. Pareto charts use the chartEx pipeline (Office 2016+); older versions of 'Microsoft Office' will display a fallback placeholder.

Office draws columns sorted by descending count plus a cumulative line on a secondary percentage axis.

Two input modes are supported:

- `aggregate = TRUE` (default): `data` is in long format (one row per observation) and Office counts occurrences of each `x` value. `y` is optional; when supplied, values are summed per category.
- `aggregate = FALSE`: `data` is already aggregated (one row per category) and `y` is the count/value column.

Usage

```
ms_paretochart(data, x, y = NULL, aggregate = TRUE)
```

Arguments

`data`	a data.frame.
`x`	category column name.
`y`	optional numeric column. With `aggregate=TRUE` and `y=NULL`, each row counts as 1.
`aggregate`	logical, see Description. Default TRUE.

Value

An `ms_chart` object (subclass `ms_paretochart`).

See Also

`chart_labels()`

Other 'Office' chart objects: `ms_areachart()`, `ms_barchart()`, `ms_boxplotchart()`, `ms_bubblechart()`, `ms_chart_combine()`, `ms_funnelchart()`, `ms_histogramchart()`, `ms_linechart()`, `ms_piechart()`, `ms_radarchart()`, `ms_scatterchart()`, `ms_stockchart()`, `ms_sunburstchart()`, `ms_treemapchart()`, `ms_waterfallchart()`

Examples

```
library(officer)

set.seed(1)
dat <- data.frame(
  defect = sample(c("A", "B", "C", "D"), 50, replace = TRUE,
                  prob = c(0.5, 0.25, 0.15, 0.1))
)
pa <- ms_paretochart(dat, x = "defect")
```

```
doc <- read_pptx()
doc <- add_slide(doc)
doc <- ph_with(doc, pa, location = ph_location_fullsize())
print(doc, target = tempfile(fileext = ".pptx"))
```

`ms_piechart` *Piechart object*

Description

Creation of a piechart object that can be inserted in a 'Microsoft' document.

Pie charts show the proportion of each category as a slice of a circle. Doughnut charts are similar but have a hole in the centre. Use `chart_settings(x, hole_size = ...)` to control the hole size: 0 produces a pie chart, values above 0 produce a doughnut chart.

Data must be pre-aggregated: one row per slice, no grouping column.

Usage

```
ms_piechart(data, x, y, labels = NULL)
```

Arguments

`data`	a data.frame
`x`	column name for categories (slices).
`y`	column name for values (slice sizes).
`labels`	column names of columns to be used as custom data labels displayed next to data points (not axis labels). Optional. If more than one name is provided, only the first one will be used as a label, but all labels (transposed if a group is used) will be available in the Excel file associated with the chart.

Value

An `ms_chart` object.

See Also

`chart_settings()`, `chart_data_labels()`, `chart_theme()`, `chart_labels()`

Other 'Office' chart objects: `ms_areachart()`, `ms_barchart()`, `ms_boxplotchart()`, `ms_bubblechart()`, `ms_chart_combine()`, `ms_funnelchart()`, `ms_histogramchart()`, `ms_linechart()`, `ms_paretochart()`, `ms_radarchart()`, `ms_scatterchart()`, `ms_stockchart()`, `ms_sunburstchart()`, `ms_treemapchart()`, `ms_waterfallchart()`

Examples

```
library(officer)
library(mschart)

dat <- data.frame(
  browser = c("Chrome", "Firefox", "Safari", "Edge", "Other"),
  value = c(64, 12, 8, 5, 11)
)

# Pie chart
pie <- ms_piechart(data = dat, x = "browser", y = "value")
pie <- chart_labels(pie, title = "Browser share")

# Doughnut chart
donut <- ms_piechart(data = dat, x = "browser", y = "value")
donut <- chart_settings(donut, hole_size = 50)
donut <- chart_labels(donut, title = "Browser share (donut)")
```

---

ms_radarchart *Radarchart object*

Description

Creation of a radar (spider) chart object that can be inserted in a 'Microsoft' document.

Usage

```
ms_radarchart(data, x, y, group = NULL, labels = NULL, asis = FALSE)
```

Arguments

data	a data.frame
x	column name for x values.
y	column name for y values.
group	grouping column name used to split data into series. Optional.
labels	column names of columns to be used as custom data labels displayed next to data points (not axis labels). Optional. If more than one name is provided, only the first one will be used as a label, but all labels (transposed if a group is used) will be available in the Excel file associated with the chart.
asis	logical parameter defaulting to FALSE. When FALSE, the data is reshaped internally so that each series becomes a separate column. When TRUE, the data is used as-is and must already have one column for categories and one column per series, and y accepts a vector of series column names. asis describes the *input shape* read by the constructor. Not to be confused with the write_data argument of sheet_add_drawing.ms_chart(), which controls whether mschart writes the chart's data into an Excel sheet at embed time. The two are independent.

Value

An ms_chart object.

See Also

chart_settings(), chart_ax_x(), chart_ax_y(), chart_data_labels(), chart_theme(), chart_labels()

Other 'Office' chart objects: ms_areachart(), ms_barchart(), ms_boxplotchart(), ms_bubblechart(), ms_chart_combine(), ms_funnelchart(), ms_histogramchart(), ms_linechart(), ms_paretochart(), ms_piechart(), ms_scatterchart(), ms_stockchart(), ms_sunburstchart(), ms_treemapchart(), ms_waterfallchart()

Examples

```
library(officer)

dat <- data.frame(
  axis = c("Sales", "Marketing", "Dev", "Support", "HR"),
  s1 = c(4, 3, 5, 2, 4),
  s2 = c(3, 5, 2, 4, 3)
)
dat_long <- data.frame(
  axis = rep(dat$axis, 2),
  value = c(dat$s1, dat$s2),
  group = rep(c("Team A", "Team B"), each = 5)
)

radar <- ms_radarchart(
  data = dat_long, x = "axis",
  y = "value", group = "group"
)
radar
```

---

ms_scatterchart *Scatterchart object*

Description

Creation of a scatterchart object that can be inserted in a 'Microsoft' document.

Usage

```
ms_scatterchart(data, x, y, group = NULL, labels = NULL, asis = FALSE)
```

Arguments

data	a data.frame
x	column name for x values.
y	column name for y values.

group	grouping column name used to split data into series. Optional.
labels	column names of columns to be used as custom data labels displayed next to data points (not axis labels). Optional. If more than one name is provided, only the first one will be used as a label, but all labels (transposed if a group is used) will be available in the Excel file associated with the chart.
asis	logical parameter defaulting to FALSE. When FALSE, the data is reshaped internally so that each series becomes a separate column. When TRUE, the data is used as-is and must already have one column for categories and one column per series, and `y` accepts a vector of series column names. `asis` describes the *input shape* read by the constructor. Not to be confused with the `write_data` argument of `sheet_add_drawing.ms_chart()`, which controls whether `mschart` writes the chart's data into an Excel sheet at embed time. The two are independent.

Value

An `ms_chart` object.

Illustrations

See Also

`chart_settings()`, `chart_ax_x()`, `chart_ax_y()`, `chart_data_labels()`, `chart_theme()`, `chart_labels()`

Other 'Office' chart objects: `ms_areachart()`, `ms_barchart()`, `ms_boxplotchart()`, `ms_bubblechart()`, `ms_chart_combine()`, `ms_funnelchart()`, `ms_histogramchart()`, `ms_linechart()`, `ms_paretochart()`, `ms_piechart()`, `ms_radarchart()`, `ms_stockchart()`, `ms_sunburstchart()`, `ms_treemapchart()`, `ms_waterfallchart()`

Examples

```
library(officer)
# example chart_01 -------
chart_01 <- ms_scatterchart(
  data = mtcars, x = "disp",
  y = "drat"
)
chart_01 <- chart_settings(chart_01, scatterstyle = "marker")


# example chart_02 -------
chart_02 <- ms_scatterchart(
  data = iris, x = "Sepal.Length",
  y = "Petal.Length", group = "Species"
)
chart_02 <- chart_settings(chart_02, scatterstyle = "marker")
```

`ms_stockchart` *Stockchart object*

Description

Creation of a stock chart object that can be inserted in a 'Microsoft' document. When open is omitted the chart is a High-Low-Close chart. When open is provided it becomes an Open-High-Low-Close chart with up/down bars (candlestick).

Usage

```
ms_stockchart(data, x, open = NULL, high, low, close)
```

Arguments

`data`	a data.frame
`x`	column name for categories (typically dates)
`open`	column name for open values (optional, enables OHLC mode)
`high`	column name for high values
`low`	column name for low values
`close`	column name for close values

Value

An `ms_chart` object.

See Also

`chart_settings()`, `chart_ax_x()`, `chart_ax_y()`, `chart_theme()`, `chart_labels()`

Other 'Office' chart objects: `ms_areachart()`, `ms_barchart()`, `ms_boxplotchart()`, `ms_bubblechart()`, `ms_chart_combine()`, `ms_funnelchart()`, `ms_histogramchart()`, `ms_linechart()`, `ms_paretochart()`, `ms_piechart()`, `ms_radarchart()`, `ms_scatterchart()`, `ms_sunburstchart()`, `ms_treemapchart()`, `ms_waterfallchart()`

Examples

```
library(officer)

dat <- data.frame(
  date = as.Date("2024-01-01") + 0:4,
  open = c(44, 25, 38, 50, 34),
  high = c(55, 57, 57, 58, 58),
  low = c(11, 12, 13, 11, 25),
  close = c(32, 35, 34, 35, 43)
)

# HLC chart
```

```
stock_hlc <- ms_stockchart(
  data = dat, x = "date",
  high = "high", low = "low", close = "close"
)
stock_hlc

# OHLC chart (candlestick)
stock_ohlc <- ms_stockchart(
  data = dat, x = "date",
  open = "open", high = "high",
  low = "low", close = "close"
)
stock_ohlc
```

`ms_sunburstchart` *Sunburst chart object*

Description

Creation of a sunburst chart object that can be inserted in a 'Microsoft' document. Sunburst charts use the chartEx pipeline (Office 2016+); older versions of 'Microsoft Office' will display a fallback placeholder.

Data is hierarchical: one column per level (parent to leaf, left to right) and one numeric column for the leaf values. Leaf values propagate to parent rings as sums.

Usage

```
ms_sunburstchart(data, path, value, labels = NULL)
```

Arguments

`data`	a data.frame.
`path`	character vector of column names defining the hierarchy, from outermost (root) to innermost (leaf).
`value`	column name for the numeric leaf values.
`labels`	unused for now; reserved for future custom data label columns.

Value

An `ms_chart` object (subclass `ms_sunburstchart`).

Per-leaf coloring

Same caveat as `ms_treemapchart()`: per-leaf colors via `chart_data_fill()` only work reliably with `length(path) == 1`.

See Also

chart_labels(), ms_treemapchart()

Other 'Office' chart objects: ms_areachart(), ms_barchart(), ms_boxplotchart(), ms_bubblechart(), ms_chart_combine(), ms_funnelchart(), ms_histogramchart(), ms_linechart(), ms_paretochart(), ms_piechart(), ms_radarchart(), ms_scatterchart(), ms_stockchart(), ms_treemapchart(), ms_waterfallchart()

Examples

```
library(officer)

dat <- data.frame(
  region  = c("EU", "EU", "EU", "AM", "AM"),
  country = c("FR", "FR", "DE", "US", "US"),
  city    = c("Paris", "Lyon", "Berlin", "NYC", "LA"),
  value   = c(10, 5, 12, 20, 8),
  stringsAsFactors = FALSE
)
sb <- ms_sunburstchart(
  data = dat, path = c("region", "country", "city"), value = "value"
)

doc <- read_pptx()
doc <- add_slide(doc)
doc <- ph_with(doc, sb, location = ph_location_fullsize())
print(doc, target = tempfile(fileext = ".pptx"))
```

---

ms_treemapchart *Treemap chart object*

Description

Creation of a treemap chart object that can be inserted in a 'Microsoft' document. Treemap charts use the chartEx pipeline (Office 2016+); older versions of 'Microsoft Office' will display a fallback placeholder.

Data is hierarchical: one column per level (parent to leaf, left to right) and one numeric column for the leaf values.

Usage

```
ms_treemapchart(data, path, value, labels = NULL)
```

Arguments

data	a data.frame.
path	character vector of column names defining the hierarchy, from outermost (root) to innermost (leaf).
value	column name for the numeric leaf values.
labels	unused for now; reserved for future custom data label columns.

Value

An ms_chart object (subclass ms_treemapchart).

Per-leaf coloring with chart_data_fill()

Per-leaf coloring via chart_data_fill() with a named vector works correctly on flat treemaps (length(path) == 1). With a hierarchy (length(path) >= 2), PowerPoint silently re-maps <cx:dataPt> indices in a way that drops idx="0" and applies the last specified color to remaining leaves. This is a PowerPoint rendering limitation (also reproducible from Excel-generated chartEx files); the mschart XML output is conformant. For hierarchical treemaps, prefer a single fill color via chart_data_fill(x, "#HEX").

See Also

chart_labels(), chart_theme()

Other 'Office' chart objects: ms_areachart(), ms_barchart(), ms_boxplotchart(), ms_bubblechart(), ms_chart_combine(), ms_funnelchart(), ms_histogramchart(), ms_linechart(), ms_paretochart(), ms_piechart(), ms_radarchart(), ms_scatterchart(), ms_stockchart(), ms_sunburstchart(), ms_waterfallchart()

Examples

```
library(officer)

dat <- data.frame(
  region  = c("EU", "EU", "EU", "AM", "AM"),
  country = c("FR", "FR", "DE", "US", "US"),
  city    = c("Paris", "Lyon", "Berlin", "NYC", "LA"),
  value   = c(10, 5, 12, 20, 8),
  stringsAsFactors = FALSE
)
tm <- ms_treemapchart(
  data = dat, path = c("region", "country", "city"), value = "value"
)
tm <- chart_labels(tm, title = "Sales by region")

doc <- read_pptx()
doc <- add_slide(doc)
doc <- ph_with(doc, tm, location = ph_location_fullsize())
print(doc, target = tempfile(fileext = ".pptx"))
```

---

ms_waterfallchart *Waterfall chart object*

Description

Creation of a waterfall chart object that can be inserted in a 'Microsoft' document. Waterfall charts use the chartEx pipeline (Office 2016+); older versions of 'Microsoft Office' will display a fallback placeholder.

Each row is one bar. Positive values rise, negative values fall. Categories listed in `subtotals` are rendered as absolute totals (typical for "Start", intermediate totals, and "End" bars).

Usage

```
ms_waterfallchart(data, x, y, subtotals = NULL)
```

Arguments

`data`	a data.frame.
`x`	category column name.
`y`	numeric value column name. Signed values: positive = up, negative = down. For subtotal rows, the value should be the running total at that point.
`subtotals`	integer vector of 1-based row indices that should be rendered as subtotal/total bars. Defaults to none.

Value

An `ms_chart` object (subclass `ms_waterfallchart`).

See Also

`chart_labels()`

Other 'Office' chart objects: `ms_areachart()`, `ms_barchart()`, `ms_boxplotchart()`, `ms_bubblechart()`, `ms_chart_combine()`, `ms_funnelchart()`, `ms_histogramchart()`, `ms_linechart()`, `ms_paretochart()`, `ms_piechart()`, `ms_radarchart()`, `ms_scatterchart()`, `ms_stockchart()`, `ms_sunburstchart()`, `ms_treemapchart()`

Examples

```
library(officer)

dat <- data.frame(
  step = c("Start", "Q1", "Q2", "Q3", "End"),
  amount = c(100, 30, -20, 40, 150),
  stringsAsFactors = FALSE
)
wf <- ms_waterfallchart(
  data = dat, x = "step", y = "amount",
  subtotals = c(1, 5)
)

doc <- read_pptx()
doc <- add_slide(doc)
doc <- ph_with(doc, wf, location = ph_location_fullsize())
```

```
print(doc, target = tempfile(fileext = ".pptx"))
```

`ph_with.ms_chart` *Add a MS Chart output into a PowerPoint object*

Description

Produces a Microsoft Chart graphics output from R instructions and adds the result in a PowerPoint document object produced by `officer::read_pptx()`.

Usage

```
## S3 method for class 'ms_chart'
ph_with(x, value, location, ...)
```

Arguments

`x`	a pptx device
`value`	chart object
`location`	a location for a placeholder.
`...`	Arguments to be passed to methods.

Value

An rpptx object.

See Also

`body_add_chart()`

Examples

```
my_barchart <- ms_barchart(data = browser_data,
  x = "browser", y = "value", group = "serie")
my_barchart <- chart_settings( x = my_barchart,
  dir="vertical", grouping="clustered", gap_width = 50 )
my_barchart <- chart_ax_x( x= my_barchart,
  cross_between = 'between', major_tick_mark="out")
my_barchart <- chart_ax_y( x= my_barchart,
  cross_between = "midCat", major_tick_mark="in")

library(officer)
doc <- read_pptx()
doc <- add_slide(doc, "Title and Content", "Office Theme")
doc <- ph_with(doc, my_barchart, location = ph_location_fullsize())

fileout <- tempfile(fileext = ".pptx")
print(doc, target = fileout)
```

print.ms_chart *Print method for ms_chart*

Description

An ms_chart object cannot be rendered in R. The default printing method will only display simple information about the object. If argument preview is set to TRUE, a pptx file will be produced and opened with function browseURL.

Usage

```
## S3 method for class 'ms_chart'
print(x, preview = FALSE, ...)
```

Arguments

x	an ms_chart object.
preview	preview the chart in a PowerPoint document
...	unused

Value

No return value, called for side effects.

set_theme *Set chart theme*

Description

Modify chart theme with function set_theme.

Use mschart_theme() to create a chart theme.

Use chart_theme() to modify components of the theme of a chart.

Usage

```
set_theme(x, value)

mschart_theme(
  axis_title = fp_text(bold = TRUE, font.size = 16),
  axis_title_x = axis_title,
  axis_title_y = axis_title,
  main_title = fp_text(bold = TRUE, font.size = 20),
  legend_text = fp_text(font.size = 14),
  table_text = fp_text(bold = FALSE, font.size = 9),
```

```
  axis_text = fp_text(),
  axis_text_x = axis_text,
  axis_text_y = axis_text,
  title_rot = 0,
  title_x_rot = 0,
  title_y_rot = 270,
  axis_ticks = fp_border(color = "#99999999"),
  axis_ticks_x = axis_ticks,
  axis_ticks_y = axis_ticks,
  grid_major_line = fp_border(color = "#99999999", style = "dashed"),
  grid_major_line_x = grid_major_line,
  grid_major_line_y = grid_major_line,
  grid_minor_line = fp_border(width = 0),
  grid_minor_line_x = grid_minor_line,
  grid_minor_line_y = grid_minor_line,
  chart_background = NULL,
  chart_border = fp_border(color = "transparent"),
  plot_background = NULL,
  plot_border = fp_border(color = "transparent"),
  date_fmt = "yyyy/mm/dd",
  str_fmt = "General",
  double_fmt = "#,##0.00",
  integer_fmt = "0",
  legend_position = "b",
  legend_x = NULL,
  legend_y = NULL,
  legend_w = NULL,
  legend_h = NULL
)

chart_theme(
  x,
  axis_title_x,
  axis_title_y,
  main_title,
  legend_text,
  title_rot,
  title_x_rot,
  title_y_rot,
  axis_text_x,
  axis_text_y,
  axis_ticks_x,
  axis_ticks_y,
  grid_major_line_x,
  grid_major_line_y,
  grid_minor_line_x,
  grid_minor_line_y,
  chart_background,
```

```
  chart_border,
  plot_background,
  plot_border,
  date_fmt,
  str_fmt,
  double_fmt,
  integer_fmt,
  legend_position,
  legend_x,
  legend_y,
  legend_w,
  legend_h
)
```

Arguments

x	an ms_chart object.
value	a mschart_theme() object.
axis_title, axis_title_x, axis_title_y	axis title formatting properties (see officer::fp_text())
main_title	title formatting properties (see officer::fp_text())
legend_text	legend text formatting properties (see officer::fp_text())
table_text	table text formatting properties (see officer::fp_text())
axis_text, axis_text_x, axis_text_y	axis text formatting properties (see officer::fp_text())
title_rot, title_x_rot, title_y_rot	rotation angle
axis_ticks, axis_ticks_x, axis_ticks_y	axis ticks formatting properties (see officer::fp_border())
grid_major_line, grid_major_line_x, grid_major_line_y	major grid lines formatting properties (see officer::fp_border())
grid_minor_line, grid_minor_line_x, grid_minor_line_y	minor grid lines formatting properties (see officer::fp_border())
chart_background	chart area background fill color - single character value (e.g. "#000000" or "black")
chart_border	chart area border lines formatting properties (see officer::fp_border())
plot_background	plot area background fill color - single character value (e.g. "#000000" or "black")
plot_border	plot area border lines formatting properties (see officer::fp_border())
date_fmt	date format
str_fmt	string or factor format
double_fmt	double format

integer_fmt integer format

legend_position
it specifies the position of the legend. It should be one of 'b', 'tr', 'l', 'r', 't', 'n' (for 'none').

legend_x, legend_y, legend_w, legend_h
optional fractions between 0 and 1 to manually position and size the legend box within the chart area. Each value is a fraction of the chart width (legend_x, legend_w) or height (legend_y, legend_h). Any NULL value keeps the default automatic layout; setting at least one triggers manual placement via <c:manualLayout>.

Value

An ms_chart object.

An mschart_theme object (for mschart_theme()).

An ms_chart object (for set_theme() and chart_theme()).

See Also

ms_barchart(), ms_areachart(), ms_scatterchart(), ms_linechart()

Examples

```
library(officer)
mytheme <- mschart_theme(
  axis_title = fp_text(color = "red", font.size = 24, bold = TRUE),
  grid_major_line_y = fp_border(width = 1, color = "orange"),
  axis_ticks_y = fp_border(width = 0.4, color = "gray")
)


my_bc <- ms_barchart(
  data = browser_data, x = "browser",
  y = "value", group = "serie"
)
my_bc <- chart_settings(my_bc,
  dir = "horizontal", grouping = "stacked",
  gap_width = 150, overlap = 100
)
my_bc <- set_theme(my_bc, mytheme)

my_bc_2 <- ms_barchart(
  data = browser_data, x = "browser",
  y = "value", group = "serie"
)
my_bc_2 <- chart_theme(my_bc_2,
  grid_major_line_y = fp_border(width = 0.5, color = "cyan")
)

# Manual legend layout: place the legend in the top-right corner
# using fractions of the chart area (0 to 1).
my_bc_3 <- ms_barchart(
```

```
  data = browser_data, x = "browser",
  y = "value", group = "serie"
)
my_bc_3 <- chart_theme(my_bc_3,
  legend_position = "r",
  legend_x = 0.80, legend_y = 0.15,
  legend_w = 0.18, legend_h = 0.30
)
```

`sheet_add_drawing.ms_chart` *Add an ms_chart to an Excel sheet*

Description

Add an `ms_chart` object to a sheet in an xlsx workbook created with `officer::read_xlsx()`. The chart data is written into the sheet and the chart is displayed at the specified position.

Usage

```
## S3 method for class 'ms_chart'
sheet_add_drawing(
  x,
  value,
  sheet,
  start_col = 1L,
  start_row = 1L,
  write_data = TRUE,
  left = 1,
  top = 1,
  width = 6,
  height = 4,
  anchor = NULL,
  edit_as = c("twoCell", "oneCell", "absolute"),
  ...
)
```

Arguments

`x`	an rxlsx object (created by `officer::read_xlsx()`)
`value`	an `ms_chart` object
`sheet`	sheet name where the chart and its data will be placed. The sheet must already exist (see `officer::add_sheet()`).
`start_col`	column index where chart data will be written (default 1, i.e. column A). When `write_data = FALSE`, this is still the cell position that the chart XML will point at, but no data is written.

start_row	row index where chart data will be written (default 1). Same semantics with write_data = FALSE as for start_col.
write_data	if TRUE (the default), the chart's data_series is written into the sheet at (start_col, start_row) before the chart is added. Pass FALSE when the data is already present in the sheet (for example to avoid rewriting it when several charts share the same dataset, or when inserting a chart that references data written independently via officer::sheet_write_data()). Not to be confused with the constructor's asis argument (see ms_barchart()), which controls how the input data frame is read at construction time. The two are independent.
left, top	top-left anchor of the chart, in inches. Defaults to (1, 1). Used when anchor = NULL. Same convention as officer::sheet_add_drawing() and rvg::sheet_add_drawing.dml().
width, height	size of the chart, in inches. Defaults to 6 x 4. Used when anchor = NULL or when anchor is a single cell reference.
anchor	optional Excel cell-based anchor. Either NULL (the default; absolute placement via left/top/width/height), a single cell reference like "B2" (the chart is anchored to that cell and keeps width/height, i.e. "Move but don't size with cells"), or a range like "B2:H20" (the chart is anchored from the first cell to the second, i.e. Excel's default "Move and size with cells").
edit_as	one of "twoCell", "oneCell", "absolute". Sets the editAs attribute on <xdr:twoCellAnchor>. Ignored unless anchor is a range.
...	unused

Value

the rxlsx object (invisibly)

Examples

```
library(officer)
library(mschart)

my_chart <- ms_barchart(
  data = data.frame(
    x = c("A", "B", "C"),
    y = c(1, 3, 2),
    group = rep("serie1", 3)
  ),
  x = "x", y = "y", group = "group"
)

x <- read_xlsx()
x <- add_sheet(x, label = "chart_sheet")
x <- sheet_add_drawing(x, value = my_chart, sheet = "chart_sheet")
print(x, target = tempfile(fileext = ".xlsx"))

# Sharing one dataset between several charts on the same sheet:
# write the data once, then add each chart with write_data = FALSE.
shared <- data.frame(
  x = c("A", "B", "C"),
```

```
  y = c(1, 3, 2),
  group = rep("serie1", 3)
)
chart_a <- ms_barchart(shared, x = "x", y = "y", group = "group")
chart_b <- ms_linechart(shared, x = "x", y = "y", group = "group")

x <- read_xlsx()
x <- add_sheet(x, label = "multi")
x <- sheet_write_data(x, value = chart_a$data_series, sheet = "multi")
x <- sheet_add_drawing(x, value = chart_a, sheet = "multi",
                       write_data = FALSE,
                       left = 3, top = 0.5, width = 5, height = 3.5)
x <- sheet_add_drawing(x, value = chart_b, sheet = "multi",
                       write_data = FALSE,
                       left = 9, top = 0.5, width = 5, height = 3.5)
print(x, target = tempfile(fileext = ".xlsx"))
```

`theme_ggplot2` *Apply ggplot2 theme*

Description

A theme that approximates the style of ggplot2::theme_grey.

Usage

```
theme_ggplot2(x, base_size = 11, base_family = "Arial")
```

Arguments

`x`	a mschart object
`base_size`	base font size
`base_family`	font family

Value

a mschart object

theme_ggplot2()

Examples

```
p <- ms_scatterchart(
  data = iris, x = "Sepal.Length",
  y = "Sepal.Width", group = "Species"
)

p <- theme_ggplot2(p)
p <- chart_fill_ggplot2(p)
```

---

us_indus_prod *Index of US Industrial Production*

Description

Index of US industrial production (1985 = 100).

Format

A data frame with 256 rows and 3 variables

Details

This is a transformation of the `USProdIndex` data from package 'AER' into a simple data.frame.

Index