Package ‘rfm’

February 26, 2024

Type Package

Title Recency, Frequency and Monetary Value Analysis

Version 0.3.0

Description Tools for RFM (recency, frequency and monetary value) analysis.
Generate RFM score from both transaction and customer level data. Visualize the relationship between recency, frequency and monetary value using heatmap, histograms, bar charts and scatter plots. Includes a 'shiny' app for interactive segmentation. References:
i. Blattberg R.C., Kim BD., Neslin S.A (2008) <doi:10.1007/978-0-387-72579-6_12>.

License MIT + file LICENSE

URL https://github.com/rsquaredacademy/rfm,
https://rfm.rsquaredacademy.com/

BugReports https://github.com/rsquaredacademy/rfm/issues

Depends R (>= 3.2)

Imports dplyr, ggplot2, magrittr, RColorBrewer, rlang, scales, stats, treemapify, utils, xplorerr

Suggests cli, covr, DT, gganimate, gifski, kableExtra, knitr, png, plotly, rmarkdown, rmdformats, testthat (>= 3.0.0), vdiffr

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Config/testthat/edition 3

NeedsCompilation no

Author Aravind Hebbali [aut, cre] (<https://orcid.org/0000-0001-9220-9669>)

Maintainer Aravind Hebbali <hebbali.aravind@gmail.com>

Repository CRAN

Date/Publication 2024-02-26 09:20:02 UTC

R topics documented:

`rfm_barchart_data` *Bar chart data*

Description

Data for generating bar charts.

Usage

```
rfm_barchart_data(rfm_table)
```

Arguments

`rfm_table`	An object of class `rfm_table`.

Examples

```
# using transaction data
analysis_date <- as.Date('2006-12-31')
rfm_order <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# bar chart data
rfm_barchart_data(rfm_order)
```

```
# using customer data
analysis_date <- as.Date('2007-01-01')
rfm_customer <- rfm_table_customer(rfm_data_customer, customer_id,
number_of_orders, recency_days, revenue, analysis_date)

# bar chart data
rfm_barchart_data(rfm_customer)
```

rfm_create_report *RFM report*

Description

Generates a segmentation analysis report.

Usage

```
rfm_create_report(
  rfm_table,
  segments,
  interactive = FALSE,
  title = NULL,
  author = NULL,
  folder_name = NULL,
  file_name = NULL
)
```

Arguments

`rfm_table`	An object of class `rfm_table`.
`segments`	Output from `rfm_segment`.
`interactive`	If `TRUE`, uses `plotly` as the visualization engine. If `FALSE`, uses `ggplot2`.
`title`	Title of the report.
`author`	Author of the report.
`folder_name`	The output directory for the report.
`file_name`	The name of the report file.

Examples

```
## Not run:
# analysis date
analysis_date <- as.Date('2006-12-31')

# generate rfm score
rfm_result <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# segment names
segment_names <- c("Champions", "Potential Loyalist", "Loyal Customers",
                   "Promising", "New Customers", "Can't Lose Them",
                   "At Risk", "Need Attention", "About To Sleep", "Lost")

# segment intervals
recency_lower <-   c(5, 3, 2, 3, 4, 1, 1, 1, 2, 1)
recency_upper <-   c(5, 5, 4, 4, 5, 2, 2, 3, 3, 1)
frequency_lower <- c(5, 3, 2, 1, 1, 3, 2, 3, 1, 1)
frequency_upper <- c(5, 5, 4, 3, 3, 4, 5, 5, 3, 5)
monetary_lower <-  c(5, 2, 2, 3, 1, 4, 4, 3, 1, 1)
monetary_upper <-  c(5, 5, 4, 5, 5, 5, 5, 5, 4, 5)

# generate segments
segments <- rfm_segment(rfm_result, segment_names, recency_lower,
recency_upper, frequency_lower, frequency_upper, monetary_lower,
monetary_upper)

rfm_create_report(rfm_result, segments, FALSE,
"Customer Segmentation Report")

## End(Not run)
```

`rfm_data_customer` *RFM customer data*

Description

A dataset containing customer level data.

Usage

```
rfm_data_customer
```

Format

A tibble with 39,999 rows and 5 variables:

customer_id Customer id.

total_amount Total amount of all orders.

most_recent_visit Date of the most recent transaction.

number_of_purchases Total number of transactions/orders.

purchase_interval Number of days since last transaction/order. #'

first_name First name of the customer.

last_name Last name of the customer.

email email id of the customer.

`rfm_data_orders` *RFM transaction data*

Description

A dataset containing transactions of different customers.

Usage

```
rfm_data_orders
```

Format

A tibble with 49.6 rows and 3 variables:

order_date Order date.

customer_id Customer id

revenue Transaction amount.

first_name First name of the customer.

last_name Last name of the customer.

email email id of the customer.

`rfm_heatmap_data` *Heatmap data*

Description

Data for generating heatmap.

Usage

```
rfm_heatmap_data(rfm_table)
```

Arguments

`rfm_table`	An object of class `rfm_table`.

Examples

```
# using transaction data
analysis_date <- as.Date('2006-12-31')
rfm_order <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# heat map data
rfm_heatmap_data(rfm_order)

# using customer data
analysis_date <- as.Date('2007-01-01')
rfm_customer <- rfm_table_customer(rfm_data_customer, customer_id,
number_of_orders, recency_days, revenue, analysis_date)

# heat map data
rfm_heatmap_data(rfm_customer)
```

`rfm_launch_app` *Launch shiny app*

Description

Launches shiny app.

Usage

```
rfm_launch_app()
```

Examples

```
## Not run:
rfm_launch_app()

## End(Not run)
```

`rfm_plot_bar_chart` *RFM bar chart*

Description

Examine the distribution of monetary scores for the different combinations of frequency and recency scores.

Usage

```
rfm_plot_bar_chart(
  rfm_table,
  bar_color = NULL,
  xaxis_label = NULL,
  sec_xaxis_label = NULL,
  yaxis_label = NULL,
  sec_yaxis_label = NULL,
  print_plot = TRUE
)
```

Arguments

`rfm_table` An object of class `rfm_table`.

`bar_color` Color of the bars.

`xaxis_label` X axis label.

`sec_xaxis_label` Secondary x axis label.

`yaxis_label` Y axis label.

`sec_yaxis_label` Secondary y axis label.

`print_plot` logical; if `TRUE`, prints the plot else returns a plot object.

Value

Bar chart.

Deprecated Functions

`rfm_bar_chart()` has been deprecated and will be made defunct. It has been provided for compatibility with older versions only, and will be made defunct at the next release.

Instead use the replacement function `rfm_plot_bar_chart()`.

Examples

```
# using transaction data
analysis_date <- as.Date('2006-12-31')
rfm_order <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# bar chart
rfm_plot_bar_chart(rfm_order)
```

rfm_plot_heatmap *RFM heatmap*

Description

The heat map shows the average monetary value for different categories of recency and frequency scores. Higher scores of frequency and recency are characterized by higher average monetary value as indicated by the darker areas in the heatmap.

Usage

```
rfm_plot_heatmap(
  data,
  brewer_n = 5,
  brewer_name = "PuBu",
  xaxis_label = NULL,
  yaxis_label = NULL,
  plot_title = NULL,
  legend_title = NULL,
  interactive = FALSE,
  print_plot = TRUE
)
```

Arguments

data	An object of class `rfm_table`.
brewer_n	Indicates the number of colors in the palette; RColorBrewer is used for the color palette of the heatmap; check the documentation of `brewer.pal`.
brewer_name	Palette name; check the documentation of `brewer.pal`.
xaxis_label	X axis label.
yaxis_label	Y axis label.
plot_title	Title of the plot.
legend_title	Legend title.
interactive	If `TRUE`, uses `plotly` as the visualization engine. If `FALSE`, uses `ggplot2`.
print_plot	logical; if `TRUE`, prints the plot else returns a plot object.

Deprecated Functions

`rfm_heatmap()` has been deprecated and will be made defunct. It has been provided for compatibility with older versions only, and will be made defunct at the next release.

Instead use the replacement function `rfm_plot_heatmap()`.

Examples

```
# using transaction data
analysis_date <- as.Date('2006-12-31')
rfm_order <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# heat map
# ggplot2
rfm_plot_heatmap(rfm_order)

# plotly
rfm_plot_heatmap(rfm_order, interactive = TRUE)

# using customer data
analysis_date <- as.Date('2007-01-01')
rfm_customer <- rfm_table_customer(rfm_data_customer, customer_id,
number_of_orders, recency_days, revenue, analysis_date)

# heat map
rfm_plot_heatmap(rfm_customer)
```

`rfm_plot_histogram` *RFM histograms*

Description

Histograms of recency, frequency and monetary value.

Usage

```
rfm_plot_histogram(
  rfm_table,
  metric = "recency",
  hist_bins = 9,
  hist_color = NULL,
  plot_title = NULL,
  xaxis_label = NULL,
  yaxis_label = NULL,
  interactive = FALSE,
  print_plot = TRUE
)
```

Arguments

`rfm_table` An object of class `rfm_table`.

`metric` Metric to be visualized. Defaults to `"recency"`. Valid values are:

- `"recency"`

- "frequency"
- "monetary"

`hist_bins` Number of bins of the histograms.

`hist_color` Color of the histogram.

`plot_title` Title of the plot.

`xaxis_label` X axis label.

`yaxis_label` Y axis label.

`interactive` If `TRUE`, uses `plotly` as the visualization engine. If `FALSE`, uses `ggplot2`.

`print_plot` logical; if `TRUE`, prints the plot else returns a plot object.

Value

Histograms

Deprecated Functions

`rfm_histograms()` has been deprecated and will be made defunct. It has been provided for compatibility with older versions only, and will be made defunct at the next release.

Instead use the replacement function `rfm_plot_histogram()`.

Examples

```
# using transaction data
analysis_date <- as.Date('2006-12-31')
rfm_order <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# histogram
# ggplot2
rfm_plot_histogram(rfm_order, metric = "frequency")

# plotly
rfm_plot_histogram(rfm_order, metric = "frequency", interactive = TRUE)

# using customer data
analysis_date <- as.Date('2007-01-01')
rfm_customer <- rfm_table_customer(rfm_data_customer, customer_id,
number_of_orders, recency_days, revenue, analysis_date)

# histogram
rfm_plot_histogram(rfm_customer)
```

rfm_plot_median_recency *Median plots*

Description

Segment wise median recency, frequency & monetary value plot.

Usage

```
rfm_plot_median_recency(
  rfm_segment_table,
  sort = FALSE,
  ascending = FALSE,
  flip = FALSE,
  bar_color = NULL,
  plot_title = NULL,
  xaxis_label = NULL,
  yaxis_label = NULL,
  axis_label_size = 8,
  axis_label_angle = 315,
  bar_labels = TRUE,
  interactive = FALSE,
  animate = FALSE,
  print_plot = TRUE
)

rfm_plot_median_frequency(
  rfm_segment_table,
  sort = FALSE,
  ascending = FALSE,
  flip = FALSE,
  bar_color = NULL,
  plot_title = NULL,
  xaxis_label = NULL,
  yaxis_label = NULL,
  axis_label_size = 8,
  axis_label_angle = 315,
  bar_labels = TRUE,
  interactive = FALSE,
  animate = FALSE,
  print_plot = TRUE
)

rfm_plot_median_monetary(
  rfm_segment_table,
  sort = FALSE,
  ascending = FALSE,
  flip = FALSE,
  bar_color = NULL,
  plot_title = NULL,
  xaxis_label = NULL,
  yaxis_label = NULL,
  axis_label_size = 8,
  axis_label_angle = 315,
  bar_labels = TRUE,
  interactive = FALSE,
  animate = FALSE,
  print_plot = TRUE
)
```

Arguments

`rfm_segment_table`
Output from `rfm_segment`.

`sort` logical; if `TRUE`, sort metrics.

`ascending` logical; if `TRUE`, sort metrics in ascending order.

`flip` logical; if `TRUE`, creates horizontal bar plot.

`bar_color` Color of the bars.

`plot_title` Title of the plot.

`xaxis_label` X axis label.

`yaxis_label` Y axis label.

`axis_label_size`
Font size of X axis tick labels.

`axis_label_angle`
Angle of X axis tick labels.

`bar_labels` If `TRUE`, add labels to the bars. Defaults to `TRUE`.

`interactive` If `TRUE`, uses `plotly` as the visualization engine. If `FALSE`, uses `ggplot2`.

`animate` If `TRUE`, animates the bars. Defaults to `FALSE`.

`print_plot` logical; if `TRUE`, prints the plot else returns a plot object.

Examples

```
# analysis date
analysis_date <- as.Date('2006-12-31')

# generate rfm score
rfm_result <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# segment names
segment_names <- c("Champions", "Potential Loyalist", "Loyal Customers",
                   "Promising", "New Customers", "Can't Lose Them",
                   "At Risk", "Need Attention", "About To Sleep", "Lost")

# segment intervals
recency_lower <-   c(5, 3, 2, 3, 4, 1, 1, 1, 2, 1)
recency_upper <-   c(5, 5, 4, 4, 5, 2, 2, 3, 3, 1)
frequency_lower <- c(5, 3, 2, 1, 1, 3, 2, 3, 1, 1)
frequency_upper <- c(5, 5, 4, 3, 3, 4, 5, 5, 3, 5)
monetary_lower <-  c(5, 2, 2, 3, 1, 4, 4, 3, 1, 1)
monetary_upper <-  c(5, 5, 4, 5, 5, 5, 5, 5, 4, 5)

# generate segments
segments <- rfm_segment(rfm_result, segment_names, recency_lower,
recency_upper, frequency_lower, frequency_upper, monetary_lower,
monetary_upper)

# plots
# visualize median recency
rfm_plot_median_recency(segments)

# plotly
rfm_plot_median_recency(segments, interactive = TRUE)

# sort in ascending order
rfm_plot_median_recency(segments, sort = TRUE, ascending = TRUE)

# default sorting is in descending order
rfm_plot_median_recency(segments, sort = TRUE)

# horizontal bars
rfm_plot_median_recency(segments, flip = TRUE)

# median frequency
rfm_plot_median_frequency(segments)

# median monetary value
rfm_plot_median_monetary(segments)
```

rfm_plot_order_dist *Customers by orders*

Description

Visualize the distribution of customers across orders.

Usage

```
rfm_plot_order_dist(
  rfm_table,
  flip = FALSE,
  bar_color = NULL,
  plot_title = NULL,
  xaxis_label = NULL,
  yaxis_label = NULL,
  bar_label_size = 3,
  bar_labels = TRUE,
  interactive = FALSE,
  animate = FALSE,
  print_plot = TRUE
)
```

Arguments

`rfm_table`	An object of class `rfm_table`.
`flip`	logical; if `TRUE`, creates horizontal bar plot.
`bar_color`	Color of the bars.
`plot_title`	Title of the plot.
`xaxis_label`	X axis title.
`yaxis_label`	Y axis title.
`bar_label_size`	Size of bar labels.
`bar_labels`	If `TRUE`, add labels to the bars. Defaults to `TRUE`.
`interactive`	If `TRUE`, uses `plotly` as the visualization engine. If `FALSE`, uses `ggplot2`.
`animate`	If `TRUE`, animates the bars. Defaults to `FALSE`.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

Value

Bar chart.

Deprecated Functions

`rfm_order_dist()` has been deprecated and will be made defunct. It has been provided for compatibility with older versions only, and will be made defunct at the next release.

Instead use the replacement function `rfm_plot_order_dist()`.

Examples

```
# using transaction data
analysis_date <- as.Date('2006-12-31')
rfm_order <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# order distribution
rfm_plot_order_dist(rfm_order)

# horizontal bars
rfm_plot_order_dist(rfm_order, flip = TRUE)
```

```
# plotly
rfm_plot_order_dist(rfm_order, interactive = TRUE)

# using customer data
analysis_date <- as.Date('2007-01-01')
rfm_customer <- rfm_table_customer(rfm_data_customer, customer_id,
number_of_orders, recency_days, revenue, analysis_date)

# order distribution
rfm_plot_order_dist(rfm_customer)
```

`rfm_plot_revenue_dist` *Revenue distribution*

Description

Customer and revenue distribution by segments.

Usage

```
rfm_plot_revenue_dist(
  x,
  flip = FALSE,
  colors = c("#3b5bdb", "#91a7ff"),
  legend_labels = c("Revenue", "Customers"),
  plot_title = "Revenue & Customer Distribution",
  xaxis_label = NULL,
  yaxis_label = NULL,
  axis_label_size = 8,
  axis_label_angle = 315,
  bar_labels = FALSE,
  bar_label_size = 2,
  interactive = FALSE,
  animate = FALSE,
  print_plot = TRUE
)
```

Arguments

`x`	An object of class `rfm_segment_summary`.
`flip`	logical; if `TRUE`, creates horizontal bar plot.
`colors`	Bar colors.
`legend_labels`	Legend labels.
`plot_title`	Title of the plot.
`xaxis_label`	X axis label.

`yaxis_label` Y axis label.

`axis_label_size` Font size of X axis tick labels.

`axis_label_angle` Angle of X axis tick labels.

`bar_labels` If TRUE, add labels to the bars. Defaults to FALSE.

`bar_label_size` Size of bar labels.

`interactive` If TRUE, uses plotly as the visualization engine. If FALSE, uses ggplot2.

`animate` If TRUE, animates the bars. Defaults to FALSE.

`print_plot` logical; if TRUE, prints the plot else returns a plot object.

Examples

```
# analysis date
analysis_date <- as.Date('2006-12-31')

# generate rfm score
rfm_result <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# segment names
segment_names <- c("Champions", "Potential Loyalist", "Loyal Customers",
                   "Promising", "New Customers", "Can't Lose Them",
                   "At Risk", "Need Attention", "About To Sleep", "Lost")

# segment intervals
recency_lower <-   c(5, 3, 2, 3, 4, 1, 1, 1, 2, 1)
recency_upper <-   c(5, 5, 4, 4, 5, 2, 2, 3, 3, 1)
frequency_lower <- c(5, 3, 2, 1, 1, 3, 2, 3, 1, 1)
frequency_upper <- c(5, 5, 4, 3, 3, 4, 5, 5, 3, 5)
monetary_lower <-  c(5, 2, 2, 3, 1, 4, 4, 3, 1, 1)
monetary_upper <-  c(5, 5, 4, 5, 5, 5, 5, 5, 4, 5)

# generate segments
segments <- rfm_segment(rfm_result, segment_names, recency_lower,
recency_upper, frequency_lower, frequency_upper, monetary_lower,
monetary_upper)

# segment summary
segment_overview <- rfm_segment_summary(segments)

# revenue distribution
# ggplot2
rfm_plot_revenue_dist(segment_overview)

# flip
rfm_plot_revenue_dist(segment_overview, flip = TRUE)

# plotly
rfm_plot_revenue_dist(segment_overview, interactive = TRUE)
```

rfm_plot_segment *RFM Segmentation Plot*

Description

Generates tree map to visualize segments.

Usage

```
rfm_plot_segment(
  table,
  metric = "customers",
  interactive = FALSE,
  print_plot = TRUE
)
```

Arguments

table	An object of class `rfm_segment_summary`.
metric	Metric to be visualized. Defaults to `"customers"`. Valid values are: • `"customers"` • `"orders"` • `"revenue"`
interactive	If `TRUE`, uses `plotly` as the visualization engine. If `FALSE`, uses `ggplot2`.
print_plot	logical; if `TRUE`, prints the plot else returns a plot object.

Examples

```
# analysis date
analysis_date <- as.Date('2006-12-31')

# generate rfm score
rfm_result <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# segment names
segment_names <- c("Champions", "Potential Loyalist", "Loyal Customers",
                   "Promising", "New Customers", "Can't Lose Them",
                   "At Risk", "Need Attention", "About To Sleep", "Lost")

# segment intervals
recency_lower <-   c(5, 3, 2, 3, 4, 1, 1, 1, 2, 1)
recency_upper <-   c(5, 5, 4, 4, 5, 2, 2, 3, 3, 1)
frequency_lower <- c(5, 3, 2, 1, 1, 3, 2, 3, 1, 1)
frequency_upper <- c(5, 5, 4, 3, 3, 4, 5, 5, 3, 5)
monetary_lower <-  c(5, 2, 2, 3, 1, 4, 4, 3, 1, 1)
monetary_upper <-  c(5, 5, 4, 5, 5, 5, 5, 5, 4, 5)
```

```
# generate segments
segments <- rfm_segment(rfm_result, segment_names, recency_lower,
recency_upper, frequency_lower, frequency_upper, monetary_lower,
monetary_upper)

# segment summary
segment_overview <- rfm_segment_summary(segments)

# treemaps
# default metric is customers
rfm_plot_segment(segment_overview)

# treemap of orders
rfm_plot_segment(segment_overview, metric = "orders")

# plotly
rfm_plot_segment(segment_overview, metric = "revenue", interactive = TRUE)
```

rfm_plot_segment_scatter *Segment Scatter Plots*

Description

Generate scatter plots to examine the relationship between recency, frequency and monetary value.

Usage

```
rfm_plot_segment_scatter(
  segments,
  x = "monetary",
  y = "recency",
  plot_title = NULL,
  legend_title = NULL,
  xaxis_label = NULL,
  yaxis_label = NULL,
  interactive = FALSE,
  animate = FALSE,
  print_plot = TRUE
)
```

Arguments

segments	Output from rfm_segment.
x	Metric to be represented on X axis.
y	Metric to be represented on Y axis.

`plot_title` Title of the plot.

`legend_title` Title of the plot legend.

`xaxis_label` X axis label.

`yaxis_label` Y axis label.

`interactive` If `TRUE`, uses `plotly` as the visualization engine. If `FALSE`, uses `ggplot2`.

`animate` If `TRUE`, animates the bars. Defaults to `FALSE`.

`print_plot` logical; if `TRUE`, prints the plot else returns a plot object.

Value

Scatter plot.

Examples

```
# analysis date
analysis_date <- as.Date('2006-12-31')

# generate rfm score
rfm_result <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# segment names
segment_names <- c("Champions", "Potential Loyalist", "Loyal Customers",
                   "Promising", "New Customers", "Can't Lose Them",
                   "At Risk", "Need Attention", "About To Sleep", "Lost")

# segment intervals
recency_lower <-   c(5, 3, 2, 3, 4, 1, 1, 1, 2, 1)
recency_upper <-   c(5, 5, 4, 4, 5, 2, 2, 3, 3, 1)
frequency_lower <- c(5, 3, 2, 1, 1, 3, 2, 3, 1, 1)
frequency_upper <- c(5, 5, 4, 3, 3, 4, 5, 5, 3, 5)
monetary_lower <-  c(5, 2, 2, 3, 1, 4, 4, 3, 1, 1)
monetary_upper <-  c(5, 5, 4, 5, 5, 5, 5, 5, 4, 5)

# generate segments
segments <- rfm_segment(rfm_result, segment_names, recency_lower,
recency_upper, frequency_lower, frequency_upper, monetary_lower,
monetary_upper)

# visualize
# ggplot2
rfm_plot_segment_scatter(segments, "monetary", "recency")

# plotly
rfm_plot_segment_scatter(segments, "monetary", "recency", interactive = TRUE)
```

rfm_plot_segment_summary *Visulaize segment summary*

Description

Generates plots for customers, orders, revenue and average order value for each segment.

Usage

```
rfm_plot_segment_summary(
  x,
  metric = NULL,
  sort = FALSE,
  ascending = FALSE,
  flip = FALSE,
  bar_color = NULL,
  plot_title = NULL,
  xaxis_label = NULL,
  yaxis_label = NULL,
  axis_label_size = 8,
  axis_label_angle = 315,
  bar_labels = TRUE,
  interactive = FALSE,
  animate = FALSE,
  print_plot = TRUE
)
```

Arguments

`x`	An object of class `rfm_segment_summary`.
`metric`	Metric to be visualized. Defaults to `"customers"`. Valid values are: • `"customers"` • `"orders"` • `"revenue"` • `"aov"`
`sort`	logical; if `TRUE`, sort metrics.
`ascending`	logical; if `TRUE`, sort metrics in ascending order.
`flip`	logical; if `TRUE`, creates horizontal bar plot.
`bar_color`	Color of the bars.
`plot_title`	Title of the plot.
`xaxis_label`	X axis label.
`yaxis_label`	Y axis label.

axis_label_size
: Font size of X axis tick labels.

axis_label_angle
: Angle of X axis tick labels.

bar_labels
: If TRUE, add labels to the bars. Defaults to TRUE.

interactive
: If TRUE, uses plotly as the visualization engine. If FALSE, uses ggplot2.

animate
: If TRUE, animates the bars. Defaults to FALSE.

print_plot
: logical; if TRUE, prints the plot else returns a plot object.

Examples

```
# analysis date
analysis_date <- as.Date('2006-12-31')

# generate rfm score
rfm_result <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# segment names
segment_names <- c("Champions", "Potential Loyalist", "Loyal Customers",
                   "Promising", "New Customers", "Can't Lose Them",
                   "At Risk", "Need Attention", "About To Sleep", "Lost")

# segment intervals
recency_lower <-   c(5, 3, 2, 3, 4, 1, 1, 1, 2, 1)
recency_upper <-   c(5, 5, 4, 4, 5, 2, 2, 3, 3, 1)
frequency_lower <- c(5, 3, 2, 1, 1, 3, 2, 3, 1, 1)
frequency_upper <- c(5, 5, 4, 3, 3, 4, 5, 5, 3, 5)
monetary_lower <-  c(5, 2, 2, 3, 1, 4, 4, 3, 1, 1)
monetary_upper <-  c(5, 5, 4, 5, 5, 5, 5, 5, 4, 5)

# generate segments
segments <- rfm_segment(rfm_result, segment_names, recency_lower,
recency_upper, frequency_lower, frequency_upper, monetary_lower,
monetary_upper)

# segment summary
segment_overview <- rfm_segment_summary(segments)

# plot segment summary
# summarize metric for all segments
# ggplot2
rfm_plot_segment_summary(segment_overview)

# plotly
rfm_plot_segment_summary(segment_overview, interactive = TRUE)

# select metric to be visualized
rfm_plot_segment_summary(segment_overview, metric = "orders")

# sort the metric in ascending order
```

```
rfm_plot_segment_summary(segment_overview, metric = "orders", sort = TRUE,
  ascending = TRUE)

# default sorting is in descending order
rfm_plot_segment_summary(segment_overview, metric = "orders", sort = TRUE)

# horizontal bars
rfm_plot_segment_summary(segment_overview, metric = "orders", flip = TRUE)
```

rfm_rm_plot *RFM Scatter plot*

Description

Examine the relationship between recency, frequency and monetary values.

Usage

```
rfm_rm_plot(
  segments,
  xaxis_label = NULL,
  yaxis_label = NULL,
  plot_title = NULL,
  print_plot = TRUE
)

rfm_fm_plot(
  segments,
  xaxis_label = NULL,
  yaxis_label = NULL,
  plot_title = NULL,
  print_plot = TRUE
)

rfm_rf_plot(
  segments,
  xaxis_label = NULL,
  yaxis_label = NULL,
  plot_title = NULL,
  print_plot = TRUE
)
```

Arguments

`segments`	Output from `rfm_segment`.
`xaxis_label`	X axis label.

`yaxis_label`	Y axis label.
`plot_title`	Title of the plot.
`print_plot`	logical; if `TRUE`, prints the plot else returns a plot object.

Value

Scatter plot.

Deprecated Functions

`rfm_rm_plot()`, `rfm_fm_plot()` and `rfm_rf_plot()` have been deprecated and will be made defunct. These functions have been provided for compatibility with older versions only, and will be made defunct at the next release. Instead use the replacement function `rfm_plot_segment_scatter()`.

`rfm_segment` *Segmentation*

Description

Create segments based on recency, frequency and monetary scores.

Usage

```
rfm_segment(
  data,
  segment_names = NULL,
  recency_lower = NULL,
  recency_upper = NULL,
  frequency_lower = NULL,
  frequency_upper = NULL,
  monetary_lower = NULL,
  monetary_upper = NULL
)
```

Arguments

`data`	An object of class `rfm_table`.
`segment_names`	Names of the segments.
`recency_lower`	Lower boundary for recency score.
`recency_upper`	Upper boundary for recency score.
`frequency_lower`	Lower boundary for frequency score.
`frequency_upper`	Upper boundary for frequency score.
`monetary_lower`	Lower boundary for monetary score.
`monetary_upper`	Upper boundary for monetary score.

Examples

```
# analysis date
analysis_date <- as.Date('2006-12-31')

# generate rfm score
rfm_result <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# segment names
segment_names <- c("Champions", "Potential Loyalist", "Loyal Customers",
                   "Promising", "New Customers", "Can't Lose Them",
                   "At Risk", "Need Attention", "About To Sleep", "Lost")

# segment intervals
recency_lower <-   c(5, 3, 2, 3, 4, 1, 1, 1, 2, 1)
recency_upper <-   c(5, 5, 4, 4, 5, 2, 2, 3, 3, 1)
frequency_lower <- c(5, 3, 2, 1, 1, 3, 2, 3, 1, 1)
frequency_upper <- c(5, 5, 4, 3, 3, 4, 5, 5, 3, 5)
monetary_lower <-  c(5, 2, 2, 3, 1, 4, 4, 3, 1, 1)
monetary_upper <-  c(5, 5, 4, 5, 5, 5, 5, 5, 4, 5)

# generate segments
segments <- rfm_segment(rfm_result, segment_names, recency_lower,
recency_upper, frequency_lower, frequency_upper, monetary_lower,
monetary_upper)

segments
```

`rfm_segment_summary` *Segment summary*

Description

An overview of customer segments.

Usage

```
rfm_segment_summary(segments)
```

Arguments

`segments`	Output from `rfm_segment`.

Examples

```
# analysis date
analysis_date <- as.Date('2006-12-31')
```

```
# generate rfm score
rfm_result <- rfm_table_order(rfm_data_orders, customer_id, order_date,
revenue, analysis_date)

# segment names
segment_names <- c("Champions", "Potential Loyalist", "Loyal Customers",
                   "Promising", "New Customers", "Can't Lose Them",
                   "At Risk", "Need Attention", "About To Sleep", "Lost")

# segment intervals
recency_lower <-   c(5, 3, 2, 3, 4, 1, 1, 1, 2, 1)
recency_upper <-   c(5, 5, 4, 4, 5, 2, 2, 3, 3, 1)
frequency_lower <- c(5, 3, 2, 1, 1, 3, 2, 3, 1, 1)
frequency_upper <- c(5, 5, 4, 3, 3, 4, 5, 5, 3, 5)
monetary_lower <-  c(5, 2, 2, 3, 1, 4, 4, 3, 1, 1)
monetary_upper <-  c(5, 5, 4, 5, 5, 5, 5, 5, 4, 5)

# generate segments
segments <- rfm_segment(rfm_result, segment_names, recency_lower,
recency_upper, frequency_lower, frequency_upper, monetary_lower,
monetary_upper)

# segment summary
rfm_segment_summary(segments)
```

rfm_table_customer_2 *RFM table (customer data)*

Description

Recency, frequency, monetary and RFM score.

Usage

```
rfm_table_customer(
  data = NULL,
  customer_id = NULL,
  n_transactions = NULL,
  recency = NULL,
  total_revenue = NULL,
  analysis_date = NULL,
  recency_bins = 5,
  frequency_bins = 5,
  monetary_bins = 5,
  ...
)
```

Arguments

`data`	A `data.frame` or `tibble`.
`customer_id`	Unique id of the customer.
`n_transactions`	Number of transactions/orders.
`total_revenue`	Total revenue from the customer.
`analysis_date`	Date of analysis.
`recency_bins`	Number of bins for recency or custom threshold.
`frequency_bins`	Number of bins for frequency or custom threshold.
`monetary_bins`	Number of bins for monetary or custom threshold.
`...`	Other arguments.
`recency`	Days since last visit or date of last visit.

Value

`rfm_table_order` returns a list with the following:

`rfm`	RFM table.
`analysis_date`	Date of analysis.
`frequency_bins`	Number of bins used for frequency score.
`recency_bins`	Number of bins used for recency score.
`monetary_bins`	Number of bins used for monetary score.
`threshold`	thresholds used for generating RFM scores.

Examples

```
analysis_date <- as.Date('2007-01-01')

# data includes days since last visit
rfm_table_customer(rfm_data_customer, customer_id, number_of_orders,
recency_days, revenue, analysis_date)

# data includes last visit date
rfm_table_customer(rfm_data_customer, customer_id, number_of_orders,
most_recent_visit, revenue, analysis_date)

# access rfm table
result <- rfm_table_customer(rfm_data_customer, customer_id, number_of_orders,
recency_days, revenue, analysis_date)
result$rfm

# using custom threshold
rfm_table_customer(rfm_data_customer, customer_id, number_of_orders,
recency_days, revenue, analysis_date, recency_bins = c(115, 181, 297, 482),
frequency_bins = c(4, 5, 6, 8), monetary_bins = c(256, 382, 506, 666))
```

`rfm_table_order` *RFM table (transaction data)*

Description

Recency, frequency, monetary and RFM score.

Usage

```
rfm_table_order(
  data = NULL,
  customer_id = NULL,
  order_date = NULL,
  revenue = NULL,
  analysis_date = NULL,
  recency_bins = 5,
  frequency_bins = 5,
  monetary_bins = 5,
  ...
)
```

Arguments

`data`	A `data.frame` or `tibble`.
`customer_id`	Unique id of the customer.
`order_date`	Date of the transaction.
`revenue`	Revenue from the customer.
`analysis_date`	Date of analysis.
`recency_bins`	Number of bins for recency or custom threshold.
`frequency_bins`	Number of bins for frequency or custom threshold.
`monetary_bins`	Number of bins for monetary or custom threshold.
`...`	Other arguments.

Value

`rfm_table_order` returns a list with the following:

`rfm`	RFM table.
`analysis_date`	Date of analysis.
`frequency_bins`	Number of bins used for frequency score.
`recency_bins`	Number of bins used for recency score.
`monetary_bins`	Number of bins used for monetary score.
`threshold`	thresholds used for generating RFM scores.

Examples

```
analysis_date <- as.Date("2006-12-31")
rfm_table_order(
  rfm_data_orders, customer_id, order_date, revenue,
  analysis_date
)

# access rfm table
result <- rfm_table_order(
  rfm_data_orders, customer_id, order_date,
  revenue, analysis_date
)
result$rfm

# using custom threshold
rfm_table_order(rfm_data_orders, customer_id, order_date, revenue,
  analysis_date,
  recency_bins = c(115, 181, 297, 482), frequency_bins = c(4, 5, 6, 8),
  monetary_bins = c(256, 382, 506, 666)
)
```

Index