Package 'mediamunger'

September 29, 2015

Title Utilities for Munging Kenshoo, Social, and Display data

Version 0.0.0.9000

Author Brian Wonch

Maintainer Brian Wonch <Brian.Wonch@gmail.com>

Description Misc functions to clean, filter, and summarise common data formats from Kenshoo and social networks.

Depends R (>= 3.2.1)

License GPL (>= 2)

LazyData true

Imports lazyeval, magrittr, scales, stringr, stringi, ggplot2, lubridate, tidyr, plyr, dplyr, ReporteRs, reshape2

R topics documented:

`apply_kpi_formatting` *Apply formatting to a KPI flextable, using functions from the scales package*

Description

Apply formatting to a KPI flextable, using functions from the scales package

Usage

```
apply_kpi_formatting(flextable, actuals_cols = NULL, delta_cols = NULL,
  comma_rows = NULL, percent_rows = NULL, dollar_rows = NULL,
  cents_rows = NULL, decimal_rows = NULL)
```

Arguments

`flextable`	an object from ReporteRs::FlexTable()
`actuals_cols`	vector of indices for columns showing actual performance (not deltas)
`delta_cols`	vector of indices for columns showing pct deltas (all percents)
`comma_rows`	vector of row indices that should apply scales::comma()
`percent_rows`	vector of row indices that should apply scales::percent()
`dollar_rows`	vector of row indices that should apply scales::dollar()
`cents_rows`	vector of row indices that should apply scales::dollar() with cents
`decimal_rows`	vector of row indices that should have comma-delimited thousands and also 2 decimal places

Value

flextable

cleanNames *Use only Lowercase and Underscore in Object Names*

Description

Use only Lowercase and Underscore in Object Names

Usage

```
cleanNames(x)
```

Arguments

x	an object for which a names attribute will be meaningful

Value

The object x with names being only lowercase and underscore

See Also

setNames

Examples

```
cleanNames(iris)
```

comma_numeric_cols *Change numeric columns into strings with commas.*

Description

Change numeric columns into strings with commas.

Usage

```
comma_numeric_cols(df)
```

Arguments

df	input data frame whose numeric columns should be made comma'd strings

Value

df with numeric columns as pretty strings, using scales::comma.

Examples

```
comma_numeric_cols(mtcars)
```

`contribution_trends` *Plot contribution trends*

Description

Plot contribution trends

Usage

```
contribution_trends(df = display_revised, time_col = "week",
  campaign_col = "campaign", grouping = "site_dcm",
  campaign_string = "continuity", agg_fun = summarise_dfa,
  kpis = c("cost", "clicks", "revenue_clickthrough", "revenue_viewthrough"),
  final_week_start = report_dates$final_week_start, nweeks = 7)
```

Arguments

`df`	input data frame
`time_col`	string giving colname with time values
`campaign_col`	string giving campaign colname
`grouping`	string giving field to group_by
`campaign_string`	string to parse for identifying campaign
`agg_fun`	aggregation function, typically a dplyr::summarise() function defined elsewhere
`kpis`	vector of strings giving KPI field names
`final_week_start`	date vector of length 1
`nweeks`	number of weeks to include

Value

plot

`date_diff` *Show period-over-period reporting*

Description

Show period-over-period reporting

Usage

```
date_diff(df, date_col, key_colname = "metric", value_colname = "value",
  cols_to_keep = NULL)
```

Arguments

df	input data frame
date_col	string giving date field colname
key_colname	string giving the "key" name (see gather_)
value_colname	string giving the "value" name (see gather_)
cols_to_keep	string vector indicating other columns to keep

Value

df period-over-period and other breakdowns

---

fb_summarise *Calculate aggregate sums and ratios of facebook metrics*

Description

Calculate aggregate sums and ratios of facebook metrics

Usage

```
fb_summarise(df)
```

Arguments

df	input data frame

Value

summarized data frame

---

filter_by_pattern *Filter a data frame by values matching a pattern*

Description

Filter a data frame by values matching a pattern

Usage

```
filter_by_pattern(df, colname, pattern, exclude = FALSE, ignore.case = TRUE)
```

Arguments

df	A data frame
colname	String column name on which to filter
pattern	Regular expression pattern to match for filtering
exclude	Exclude cases matching pattern? Defaults to FALSE.
ignore.case	Boolean whether string searches should be case-insensitive

Value

a filtered data frame

Examples

```
filter_by_pattern(iris, "Species", "v.r")
```

`find_display_filenames` *Grep a chosen folder for certain types of display files*

Description

Grep a chosen folder for certain types of display files

Usage

```
find_display_filenames(folder, type)
```

Arguments

`folder`	string specifying the location of display files
`type`	one of c("adwords", "front_end", or "floodlight")

Value

string showing full path to the chosen file

`fiscal_calendar` *Table of fiscal calendar lookups from 2012-12-01 to 2016-12-31 A dataset containing attributes of dates from 2012-12-01 to 2016-12-31*

Description

Table of fiscal calendar lookups from 2012-12-01 to 2016-12-31 A dataset containing attributes of dates from 2012-12-01 to 2016-12-31

Usage

```
fiscal_calendar
```

Format

A data frame with 1820 rows and 19 variables:

- fiscal_year. character for fiscal year beginning in February
- fiscal_month. character for 3-letter month abbreviation
- fiscal_month_num. numeral character for fiscal month, February is 1
- current_date. Date format, primary key
- day_num. character for day number of fiscal calendar beginning in February
- current_week_num. character for fiscal week numeral beginning in February
- current_start_of_week. date for the Sunday starting the current_date
- previous_year_start_of_week. date for the Sunday starting the current_date - 364
- fiscal_quarter. character for the fiscal quarter 1 through 4, beginning February
- previous_year_date. current_date - 364
- day_of_week_num. character of numerals 1 through 7, Sunday is 1
- day_of_week. character of 3-letter day abbreviation for current_date
- season. chr "Spring" (February through July) or "Fall" (August through January)
- previous_week_date. current_date - 7
- total_weeks_in_fiscal_month. int numer of weeks in fiscal month
- total_days_in_fiscal_month. int number of days in fiscal month
- fiscal_month_start. date of the Sunday beginning the fiscal month
- day_num_in_fiscal_month. character indicating the day number in current fiscal month
- calendar_month_start. character indicating the first date of the calendar month for current_date

---

format_deltas *Conditional formatting for a time-comparison FlexTable*

Description

Conditional formatting for a time-comparison FlexTable

Usage

```
format_deltas(flextable, data_df, delta_col_nums, cost_ratio_row_nums,
  small_delta = 0.1, big_delta = 0.3)
```

Arguments

flextable	An object from ReporteRs::FlexTable()
data_df	Data source for the flextable object
delta_col_nums	vector giving indices of columns that show percentage comparison
cost_ratio_row_nums	vector giving indices of rows that show cost_per_* metrics
small_delta	criterion for highlighting minor relative increases or decreases in decimal form. Default is 0.1
big_delta	criterion for highlighting major relative increases or decreases in decimal form. Default is 0.3

Value

flextable object with formatting applied

`get_comparison_table` *Calculate period-over-period report stacking KPIs*

Description

Calculate period-over-period report stacking KPIs

Usage

```
get_comparison_table(df, campaign_name, campaign_colname,
  group_colname = "week", agg_fun)
```

Arguments

`df`	input data frame
`campaign_name`	campaign name to report in the table
`campaign_colname`	column name indicating "campaign"
`group_colname`	string vector indicating columns to group by
`agg_fun`	summarising function

Value

filtered data frame

`get_metric_position_indices` *Grep a vector of KPI names matching a convention for formatting*

Description

Grep a vector of KPI names matching a convention for formatting

Usage

```
get_metric_position_indices(metrics_vec, type = "comma")
```

Arguments

`metrics_vec`	vector of KPIs, likely colnames from a KPI data frame
`type`	choose among c("percent", "dollar", "cost_per", "cents", "comma")

Value

vector

get_top_campaign_names

Get a vector of top campaign names in a sorted dataframe

Description

Get a vector of top campaign names in a sorted dataframe

Usage

```
get_top_campaign_names(df, week_num, n = 3, sort_on = "cost",
  week_lookup_colname = "week", aggregate_function)
```

Arguments

df	data frame
week_num	fiscal week number for filtering
n	number of top campaigns to select (default is 3)
sort_on	column to sort descending
week_lookup_colname	column name indicating "week"
aggregate_function	summarising function

Value

data frame

get_week_comparison_lookups

Filter a lookup table to relevant dates for WoW and YoY comparisons

Description

Filter a lookup table to relevant dates for WoW and YoY comparisons

Usage

```
get_week_comparison_lookups(fiscal_calendar, final_week_num, this_fiscal_year)
```

Arguments

fiscal_calendar	dataframe lookup table
final_week_num	fiscal week number as character
this_fiscal_year	fiscal year as character

Value

filtered data frame with date information

join_dcm_frames *Merge floodlight and basic-report data frames*

Description

Merge floodlight and basic-report data frames

Usage

```
join_dcm_frames(front_end_df, floodlight_df)
```

Arguments

front_end_df	dataframe with "Basic" DoubleClick report data
floodlight_df	dataframe with Basic Floodlight report data

Value

merged data frame

list_fiscal_details *Produce a list of common fiscal information, using a lookup table*

Description

Produce a list of common fiscal information, using a lookup table

Usage

```
list_fiscal_details(fiscal_calendar,
  week_num_lookup_colname = "current_week_num",
  date_lookup_colname = "current_date",
  fiscal_year_lookup_colname = "fiscal_year", current_date = NULL)
```

Arguments

fiscal_calendar	data frame with lookups. An example is in data(fiscal_calendar)
week_num_lookup_colname	column name for fiscal week number in fiscal_calendar
date_lookup_colname	column name indicating date in fiscal_calendar
fiscal_year_lookup_colname	column name indicating fiscal year in fiscal_calendar
current_date	defaults to today.

Value

list of common fiscal date info

Examples

```
list_fiscal_details(fiscal_calendar)
```

---

load_kenshoo_ftp *Load a Kenshoo report from an FTP location*

Description

Load a Kenshoo CSV report from an FTP location

Usage

```
load_kenshoo_ftp(report_name, ftp_address = "ftp.kenshoo.com",
  is.csv = TRUE, username = NULL, password = NULL, save.creds = FALSE,
  creds.file = NULL)
```

Arguments

report_name	String of the report filename. ".csv" is appended if you exclude it.
ftp_address	String with the domain name. Default is "ftp.kenshoo.com"
is.csv	Boolean if the file is csv.
username	FTP site username.
password	FTP site password. Can be left as NULL if this function is run in the current R session or if the Kenshoo.FTP.Creds list is saved on disk
save.creds	Boolean to save Kenshoo.FTP.Creds to disk for next time.
creds.file	String to supply location of already-saved Kenshoo.FTP.Creds. If NULL, the function will check "~/Kenshoo.FTP.Creds"

Value

A data frame for the specified file.

---

proper *Convert a string to proper case*

Description

Convert a string to proper case

Usage

```
proper(str)
```

Arguments

str	a character string or vector of strings

Value

string in proper case

Examples

```
proper("abc")
```

---

proper_col *Make a "metric" column proper*

Description

Make a "metric" column proper

Usage

```
proper_col(df, colname, levels_vec = NULL)
```

Arguments

df	a dataframe with a column named "metric"
colname	String giving the name of the column whose values should be made characters in proper case.
levels_vec	an optional vector of strings giving levels of a factor.

Value

data frame with "metric" column in proper case

Examples

```
proper_col(CO2, "Treatment")
```

---

proper_names *Convert column names to proper case and convert underscore to spaces*

Description

Convert column names to proper case and convert underscore to spaces

Usage

```
proper_names(df, underscore_to_spaces = TRUE)
```

Arguments

df	a dataframe with column names
underscore_to_spaces	boolean on whether to convert underscore to space in column names

Value

dataframe with adjusted column names

Examples

```
proper_names(mtcars)
```

---

read_dcm_file *Read a DoubleClick file accounting for extra header rows and totals column*

Description

Read a DoubleClick file accounting for extra header rows and totals column

Usage

```
read_dcm_file(filename, skip = 0)
```

Arguments

filename	a character string giving the path to a DoubleClick report CSV
skip	number of rows to skip if known. If set at 0, will search for "Report Fields" within first 100 rows.

Value

a data frame

---

return_match *Return matched regexp group*

Description

Return matched regexp group

Usage

```
return_match(string, pattern)
```

Arguments

string	a string or vector of strings
pattern	a regexp string including parens to find a group

Value

a vector isolating the matching group, or NA where match not found

Examples

```
return_match(row.names(mtcars), "(^[^\\s]+)")
```

segment_trend_plot *Trended geom_barplot with segmented trended geom_barplots on top*

Description

Trended geom_barplot with segmented trended geom_barplots on top

Usage

```
segment_trend_plot(df, trend_col, segmentation_col)
```

Arguments

df	input data frame
trend_col	column with top-level trends
segmentation_col	column for lower_level segmentation

Value

printed plot

start_of_month *First date of calendar month*

Description

First date of calendar month

Usage

```
start_of_month(date)
```

Arguments

date	A 'Date' object, or character string in the format "%Y-%m-%d" or "%Y/%m/%d"

Value

A Date object that is the first date of the respective calendar month.

Examples

```
start_of_month("2014/02/14")
```

strip_creative_size *Display strip creative size*

Description

Given an input vector, strips common elements designating creative sizes for display

Usage

```
strip_creative_size(colname)
```

Arguments

colname	field whose elements should be stripped of creative size

Value

a vector stripped of creative size

summarise_dfa *Calculate common display aggregates on a DoubleClick data frame*

Description

Calculate common display aggregates on a DoubleClick data frame

Usage

```
summarise_dfa(df)
```

Arguments

df	data frame with DoubleClick report data

Value

summarised tbl_df

summarise_kenshoo_metrics *Summarise Kenshoo Metrics*

Description

Summarise Kenshoo Metrics

Usage

```
summarise_kenshoo_metrics(df)
```

Arguments

df	A data frame with metrics following Kenshoo conventions

Value

An aggregated data frame with metrics following Kenshoo conventions, aggregated.

theme_nice *ggplot2 theme with pretty default settings*

Description

ggplot2 theme with pretty default settings

Usage

```
theme_nice(base_size = 20, base_family = "")
```

Arguments

base_size	reference font size. Default is 20.
base_family	reference font family

Value

set of modified ggplot2 theme elements

Examples

```
ggplot(mtcars, aes(x=mpg, y=wt)) + geom_point() + theme_nice(base_size=12)
```

`time_comparison_flextable` *Create a period comparison table and conditionally format as a FlexTable*

Description

Create a period comparison table and conditionally format as a FlexTable

Usage

```
time_comparison_flextable(week_comparison_df, final_week, previous_week,
  campaign_to_filter, campaign_col = "campaign", group_colname = "week",
  last_year_final_week = NULL, agg_fun)
```

Arguments

`week_comparison_df`	performance dataframe filtered to relevant periods for time comparison (e.g. just last-week, previous-week, and last-week-last-year)
`final_week`	string identifying the most recent period
`previous_week`	string identifying the previous period
`campaign_to_filter`	string or regexp indicating campaign to filter on
`campaign_col`	string identifying the "campaign" column for filtering
`group_colname`	period colname, typically "week"
`last_year_final_week`	optional string identifying a third period (e.g. for year-over-year reporting)
`agg_fun`	summarise function

Value

formatted flextable

Index