Package 'AutoAds'

May 6, 2026

Type Package

Title Advertisement Metrics Calculation

Version 0.1.0

Description Calculations of the most common metrics of automated advertisement and plotting of them with trend and forecast. Calculations and description of metrics is taken from different RTB platforms support documentation. Plotting and forecasting is based on packages 'forecast', described in Rob J Hyndman and George Athanasopoulos (2021) ``Forecasting: Principles and Practice" <https://otexts.com/fpp3/> and Rob J Hyndman et al ``Documentation for 'forecast'" (2003) <https://pkg.robjhyndman.com/forecast/>, and 'ggplot2', described in Hadley Wickham et al ``Documentation for 'ggplot2'" (2015) <https://ggplot2.tidyverse.org/>, and Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen (2015) ``ggplot2: Elegant Graphics for Data Analysis" <https://ggplot2-book.org/>.

License MIT + file LICENSE

Encoding UTF-8

Imports forecast, ggplot2, rlang

NeedsCompilation no

Author Ivan Nemtsev [aut, cre, cph]

Maintainer Ivan Nemtsev <nemtsev.v@gmail.com>

Repository CRAN

Date/Publication 2023-07-08 17:50:02 UTC

Contents

`adstats` *All metrics*

Description

A combination of functions view_percent, eCPM, CPMv, CTR, and fill_rate in one.

Usage

```
adstats(data)
```

Arguments

`data`	A dataset downloaded from excel file, has to be assigned to 'data' and include columns 'Requests', 'Impressions', 'Revenue', 'Viewable', and 'Clicks'.

Value

`view_percent`	A percentage of viewable impressions among the total number of impressions.
`eCPM`	Cost per thousand impressions of an ad, shown in the currency the original file had for 'Revenue'.
`CPMv`	Cost per thousand viewable impressions of an ad, shown in the currency the original file had for 'Revenue'.
`CTR`	A ratio of clicks to impressions.
`fill_rate`	A ratio of impressions to requests as a percentage.

Note

Dataset downloaded from excel file, has to be assigned to 'data' and include columns 'Requests', 'Impressions', 'Revenue', 'Viewable', and 'Clicks'.

Author(s)

Ivan Nemtsev

Examples

```
## The function is currently defined as
adstats <- function(data){
  data$ViewablePercent <- view_percent(data)
  data$eCPM <- eCPM(data)
  data$CPMv <- CPMv(data)
  data$CTR <- CTR(data)
  data$FillRate <- fill_rate(data)
}

## Example of use:
```

```
data <- data.frame(
 Date = c("2022-07-01", "2022-07-02", "2022-07-03", "2022-07-29", "2022-07-30", "2022-07-31"),
  Block = c("1_234", "1_234", "1_234", "1_235", "1_235", "1_235"),
  Requests = c(372234, 268816, 291224, 1928854, 1928290, 786539),
  Impressions = c(18537, 12432, 13764, 2839269, 2682648, 1114773),
  Revenue = c(13.5, 9.13, 8.85, 1669.0, 1654.0, 739.0),
  Clicks = c(1167, 720, 856, 214451, 196657, 93178),
  Viewable = c(13320, 8214, 9768, 2446884, 2243865, 1063158)
)
data <- adstats(data)
```

CPC *Cost per Click*

Description

A ratio showing how much actual clicks on the ad cost.

Usage

```
CPC(data)
```

Arguments

data	Dataset downloaded from excel file, has to be assigned to 'data' and include columns 'Clicks' and 'Revenue'.

Value

A ratio of revenue to clicks - a cost of one click.

Note

Dataset has to be named 'data' and include columns 'Clicks' and 'Revenue'.

Author(s)

Ivan Nemtsev

Examples

```
## The function is currently defined as
CPC <- function(data){
  x <- data$Clicks
  y <- data$Revenue
  round(
    y/x,
    digits=2)
}
```

```
## Example of use:
data <- data.frame(
 Date = c("2022-07-01", "2022-07-02", "2022-07-03", "2022-07-29", "2022-07-30", "2022-07-31"),
  Block = c("1_234", "1_234", "1_234", "1_235", "1_235", "1_235"),
  Requests = c(372234, 268816, 291224, 1928854, 1928290, 786539),
  Impressions = c(18537, 12432, 13764, 2839269, 2682648, 1114773),
  Revenue = c(13.5, 9.13, 8.85, 1669.0, 1654.0, 739.0),
  Clicks = c(1167, 720, 856, 214451, 196657, 93178),
  Viewable = c(13320, 8214, 9768, 2446884, 2243865, 1063158)
)
data$CPC <- CPC(data)
```

CPMv *Cost per thousand viewable impressions*

Description

Calculating CPMv based on Viewable Impressions and Revenue data. The dataset has to be named 'data' and include columns 'Viewable' and 'Revenue'.

Usage

```
CPMv(data)
```

Arguments

data	Dataset downloaded from excel file, has to be assigned to 'data' and include columns 'Viewable' and 'Revenue'.

Value

Cost per thousand viewable impressions of an ad, shown in the currency the original file had for 'Revenue'.

Note

Dataset has to be named 'data' and include columns 'Viewable' and 'Revenue'.

Author(s)

Ivan Nemtsev

Examples

```
## The function is currently defined as
CPMv <- function(data){
  x <- data$Viewable
  y <- data$Revenue
  round(
    y/x*1000,
    digits=2)
}

## Example of use:
data <- data.frame(
 Date = c("2022-07-01", "2022-07-02", "2022-07-03", "2022-07-29", "2022-07-30", "2022-07-31"),
  Block = c("1_234", "1_234", "1_234", "1_235", "1_235", "1_235"),
  Requests = c(372234, 268816, 291224, 1928854, 1928290, 786539),
  Impressions = c(18537, 12432, 13764, 2839269, 2682648, 1114773),
  Revenue = c(13.5, 9.13, 8.85, 1669.0, 1654.0, 739.0),
  Clicks = c(1167, 720, 856, 214451, 196657, 93178),
  Viewable = c(13320, 8214, 9768, 2446884, 2243865, 1063158)
)
data$CPMv <- CPMv(data)
```

CTR *Clickthrough rate*

Description

A ratio showing how often people who see your ad end up clicking on it.

Usage

```
CTR(data)
```

Arguments

data	Dataset downloaded from excel file, has to be assigned to 'data' and include columns 'Impressions' and 'Clicks'.

Value

A ratio of clicks to impressions.

Note

Dataset has to be named 'data' and include columns 'Impressions' and 'Clicks'.

Author(s)

Ivan Nemtsev

Examples

```
## The function is currently defined as
CTR <- function(data){
  x <- data$Impressions
  y <- data$Clicks
  round(
    y/x,
    digits=2)
}

## Example of use:
data <- data.frame(
 Date = c("2022-07-01", "2022-07-02", "2022-07-03", "2022-07-29", "2022-07-30", "2022-07-31"),
  Block = c("1_234", "1_234", "1_234", "1_235", "1_235", "1_235"),
  Requests = c(372234, 268816, 291224, 1928854, 1928290, 786539),
  Impressions = c(18537, 12432, 13764, 2839269, 2682648, 1114773),
  Revenue = c(13.5, 9.13, 8.85, 1669.0, 1654.0, 739.0),
  Clicks = c(1167, 720, 856, 214451, 196657, 93178),
  Viewable = c(13320, 8214, 9768, 2446884, 2243865, 1063158)
)
data$CTR <- CTR(data)
```

---

eCPM *Effective cost per thousand impressions*

Description

Calculating eCPM based on Impressions and Revenue data. The dataset has to be named 'data' and include columns 'Impressions' and 'Revenue'.

Usage

```
eCPM(data)
```

Arguments

`data`	Dataset downloaded from excel file, has to be assigned to 'data' and include columns 'Impressions' and 'Revenue'.

Value

Cost per thousand impressions of an ad, shown in the currency the original file had for 'Revenue'.

Note

Dataset has to be named 'data' and include columns 'Impressions' and 'Revenue'.

Author(s)

Ivan Nemtsev

Examples

```
## The function is currently defined as
eCPM <- function(data){
  x <- data$Impressions
  y <- data$Revenue
  round(
    y/x*1000,
    digits=2)
}

## Example of use:
data <- data.frame(
 Date = c("2022-07-01", "2022-07-02", "2022-07-03", "2022-07-29", "2022-07-30", "2022-07-31"),
  Block = c("1_234", "1_234", "1_234", "1_235", "1_235", "1_235"),
  Requests = c(372234, 268816, 291224, 1928854, 1928290, 786539),
  Impressions = c(18537, 12432, 13764, 2839269, 2682648, 1114773),
  Revenue = c(13.5, 9.13, 8.85, 1669.0, 1654.0, 739.0),
  Clicks = c(1167, 720, 856, 214451, 196657, 93178),
  Viewable = c(13320, 8214, 9768, 2446884, 2243865, 1063158)
)
data$eCPM <- eCPM(data)
```

---

fill_rate *Fill Rate/Show Rate*

Description

A percentage of returned ads that were displayed in the app to the user out of all requested ones.

Usage

```
fill_rate(data)
```

Arguments

data	Dataset downloaded from excel file, has to be assigned to 'data' and include columns 'Requests' and 'Impressions'.

Value

A ratio of impressions to requests as a percentage.

Note

Dataset has to be named 'data' and include columns 'Requests' and 'Impressions'.

Author(s)

Ivan Nemtsev

Examples

```
## The function is currently defined as
fill_rate <- function(data){
  x <- data$Requests
  y <- data$Impressions
  round(
    y/x*100,
    digits=2)
}

## Example of use:
data <- data.frame(
 Date = c("2022-07-01", "2022-07-02", "2022-07-03", "2022-07-29", "2022-07-30", "2022-07-31"),
  Block = c("1_234", "1_234", "1_234", "1_235", "1_235", "1_235"),
  Requests = c(372234, 268816, 291224, 1928854, 1928290, 786539),
  Impressions = c(18537, 12432, 13764, 2839269, 2682648, 1114773),
  Revenue = c(13.5, 9.13, 8.85, 1669.0, 1654.0, 739.0),
  Clicks = c(1167, 720, 856, 214451, 196657, 93178),
  Viewable = c(13320, 8214, 9768, 2446884, 2243865, 1063158)
)
data$FillRate <- fill_rate(data)
```

---

plot_trend_and_forecast

Plot with trend and forecast

Description

Plotting metrics of the dataset with trend and forecast, splitting them by groups of different ad blocks.

Usage

```
plot_trend_and_forecast(data, x_col, y_col, group_col)
```

Arguments

data	Dataset downloaded from excel file, has to be assigned to 'data' and include columns 'Date', 'Block', and at least one with the data needed to plot ('Requests', 'Impressions', 'Revenue', 'Viewable', 'Clicks').
x_col	Column of the dataset 'data' with the data required for the x-axis of the plot. Preferrably 'Date'.

y_col	Column of the dataset 'data' with the data required for the y-axis of the plot. Preferrably the one with the data needed to plot ('Requests', 'Impressions', 'Revenue', 'Viewable', 'Clicks').
group_col	Column of the dataset 'data' with the data required for grouping the data. Preferrably 'Block'.

Value

A plot of the chosen data with trend and forecast.

Note

This function also requires packages 'ggplot2', and 'forecast'. Dataset downloaded from excel file, has to be assigned to 'data' and include columns 'Date', 'Block', and at least one with the data needed to plot ('Requests', 'Impressions', 'Revenue', 'Viewable', 'Clicks').

Author(s)

Ivan Nemtsev

References

Hyndman, R. J., & Athanasopoulos, G. (2018). "Forecasting: Principles and Practice" - https://otexts.com/fpp3/ Documentation for 'forecast' - https://pkg.robjhyndman.com/forecast/ Documentation for 'ggplot2' - https://ggplot2.tidyverse.org/ Wickham, H. (2016). "ggplot2: Elegant Graphics for Data Analysis" - https://ggplot2-book.org/

Examples

```
## The function is currently defined as
plot_trend_and_forecast <- function(data, x_col, y_col, group_col) {
  data$Date <- as.Date(data$Date)
  data_groups <- split(data, data[[group_col]])
  for (group in names(data_groups)) {
    group_data <- data_groups[[group]]
   time_series <- ts(group_data[[y_col]], frequency = 1, start = min(group_data[[x_col]]))
    arima_model <- auto.arima(time_series)
    forecast <- forecast(arima_model, h = 6)
    forecast_dates <- seq(max(group_data[[x_col]]) + 1,
    to = max(group_data[[x_col]]) + length(forecast$mean),
    by = "day")
    forecast_data <- data.frame(Date = forecast_dates, Forecast = forecast$mean)
    print(ggplot() +
        geom_line(data = group_data, aes(x = !!rlang::sym(x_col), y = !!rlang::sym(y_col))) +
           geom_line(data = forecast_data, aes(x = Date, y = Forecast), color = "blue") +
            geom_ribbon(data = forecast_data, aes(x = Date, ymin = forecast$lower[, 2],
            ymax = forecast$upper[, 2]), fill = "blue", alpha = 0.2) +
            labs(x = x_col, y = y_col, title = group) +
            theme_minimal())
  }
}
```

```
## Example of use:
data <- data.frame(
 Date = c("2022-07-01", "2022-07-02", "2022-07-03", "2022-07-29", "2022-07-30", "2022-07-31"),
  Block = c("1_234", "1_234", "1_234", "1_235", "1_235", "1_235"),
  Requests = c(372234, 268816, 291224, 1928854, 1928290, 786539),
  Impressions = c(18537, 12432, 13764, 2839269, 2682648, 1114773),
  Revenue = c(13.5, 9.13, 8.85, 1669.0, 1654.0, 739.0),
  Clicks = c(1167, 720, 856, 214451, 196657, 93178),
  Viewable = c(13320, 8214, 9768, 2446884, 2243865, 1063158)
)
plot_trend_and_forecast(data, "Date", "Impressions", "Block")
#Any value instead of Impressions can be here
```

`view_percent` *Percentage of views*

Description

A ratio of viewable impressions to all impressions, shown as a percent.

Usage

```
view_percent(data)
```

Arguments

`data`	Dataset downloaded from excel file, has to be assigned to 'data' and include columns 'Impressions' and 'Viewable'.

Value

A percentage of viewable impressions among the total number of impressions.

Note

Dataset has to be named 'data' and include columns 'Impressions' and 'Viewable'.

Author(s)

Ivan Nemtsev

Examples

```
## The function is currently defined as
view_percent <- function(data){
  x <- data$Impressions
  y <- data$Viewable
  round(
    y/x*100,
  digits=2)
}

## Example of use:
data <- data.frame(
 Date = c("2022-07-01", "2022-07-02", "2022-07-03", "2022-07-29", "2022-07-30", "2022-07-31"),
  Block = c("1_234", "1_234", "1_234", "1_235", "1_235", "1_235"),
  Requests = c(372234, 268816, 291224, 1928854, 1928290, 786539),
  Impressions = c(18537, 12432, 13764, 2839269, 2682648, 1114773),
  Revenue = c(13.5, 9.13, 8.85, 1669.0, 1654.0, 739.0),
  Clicks = c(1167, 720, 856, 214451, 196657, 93178),
  Viewable = c(13320, 8214, 9768, 2446884, 2243865, 1063158)
)
data$ViewablePercent <- view_percent(data)
```

Index