

# Package ‘marketr’

February 10, 2020

**Title** Tidy Calculation of Marketing Metrics Plus Quick Analysis

**Version** 0.0.1

**Description** Facilitates tidy calculation of popular quantitative marketing metrics. It also includes functions for doing analysis that will help marketers and data analysts better understand the drivers and/or trends of these metrics. These metrics include Customer Experience Index <<https://go.forrester.com/analytics/cx-index/>> and Net Promoter Score <<https://www.netpromoter.com/know/>>.

**Depends** R (>= 3.5.0)

**License** CC0

**Encoding** UTF-8

**LazyData** true

**Suggests** knitr, rmarkdown, ggplot2

**VignetteBuilder** knitr

**Imports** dplyr (>= 0.8.3), magrittr (>= 1.5), tidyr (>= 1.0.0), tibble (>= 2.1.3), rlang (>= 0.4.2)

**RoxygenNote** 7.0.2

**NeedsCompilation** no

**Author** Chris Umphlett [aut, cre]

**Maintainer** Chris Umphlett <[christopher.umphlett@gmail.com](mailto:christopher.umphlett@gmail.com)>

**Repository** CRAN

**Date/Publication** 2020-02-10 15:20:02 UTC

## R topics documented:

exi_calc	2
exi_trend	3
nps_calc	4
nps_trend	5

<b>Index</b>	<b>7</b>
--------------	----------

---

`cxi_calc`*Tidy Calculation of Customer Experience Index*

---

### Description

Simplifies the calculation of Customer Experience Index (CXi) from raw survey data within the tidyverse framework.

### Usage

```
cxi_calc(survey_data, ..., cx_high = 4, cx_low = 2)
```

### Arguments

<code>survey_data</code>	Raw survey data. Needs to be one row per survey with the three CXi question responses having column names of needs, ease and emotion
<code>...</code>	optional columns by which to group the CXi calculation. There is no limit to the number of grouping variables chosen. Too many will likely result in CXi calculations that are too fragmented / based on very small survey counts.
<code>cx_high</code>	Threshold in scale where response at or above is a "high"
<code>cx_low</code>	Threshold in scale where response at or below is a "low"

### Details

Customer Experience Index is a metric created by Forrester to help companies systematically measure customer experience in a way that their research has found is connected to improving customer loyalty. More information can be found at <https://go.forrester.com/analytics/cx-index/>

The calculation across an entire sample of surveys is simple. A customer experience manager may want to calculate CXi across many different dimensions and filtering in different ways; the functions in this package utilize the tidy framework to streamline calculating CXi along as many dimensions as desired.

### Value

Data frame with CXi and survey count for each combination of the grouping variables

### Examples

```
needs <- sample(5, 100, replace = TRUE)
ease <- sample(5, 100, replace = TRUE)
emotion <- sample(5, 100, replace = TRUE)
cx_date <- rep(seq.Date(from = as.Date("2019-01-01"), to = as.Date("2019-01-10"), by = "day"), 10)
cx_group <- rep(c("a", "b", "c", "d"), 25)
df <- data.frame(needs, ease, emotion, cx_date, cx_group)
cxi_calc(df, cx_group)
```

cxi\_trend

*Tidy Calculation of Customer Experience Index trends by group***Description**

Simplifies the calculation of Customer Experience Index (CXi) trends over time from raw survey data within the tidyverse framework.

**Usage**

```
cxi_trend(
  survey_data,
  trend_var,
  ...,
  cx_high = 4,
  cx_low = 2,
  min_surveys = 1,
  avg_surveys = 0
)
```

**Arguments**

survey_data	Raw survey data. Needs to be one row per survey with the three CXi question responses having column names of needs, ease and emotion
trend_var	Column that represents an element of time, eg week number, date, month & year
...	optional columns by which to group the CXi calculation. There is no limit to the number of grouping variables chosen. Too many will likely result in CXi calculations that are too fragmented / based on very small survey counts.
cx_high	Threshold in scale where response at or above is a "high"
cx_low	Threshold in scale where response at or below is a "low"
min_surveys	Minimum surveys found in every period for each group to be included
avg_surveys	Average surveys found in every period for each group to be included

**Details**

Customer Experience Index is a metric created by Forrester to help companies systematically measure customer experience in a way that their research has found is connected to improving customer loyalty. More information can be found at <https://go.forrester.com/analytics/cx-index/>

The calculation across an entire sample of surveys is simple. A customer experience manager may want to calculate CXi across many different dimensions and filtering in different ways; the functions in this package utilize the tidy framework to streamline calculating CXi along as many dimensions as desired.

The trend version of the function allows you to specify one column as a date over which to trend the data. This allows quick filtering to eliminate groupings that fail to meet user-specified thresholds for average or minimum survey counts per time period.

The resulting data set is set up for creating faceted line plots using ggplot2.

**Value**

Data frame with CXi and survey count for each combination of the grouping variables over the time variable.

**Examples**

```
needs <- sample(5, 100, replace = TRUE)
ease <- sample(5, 100, replace = TRUE)
emotion <- sample(5, 100, replace = TRUE)
cx_date <- rep(seq.Date(from = as.Date("2019-01-01"), to = as.Date("2019-01-10"), by = "day"), 10)
cx_group <- rep(c("a", "b", "c", "d"), 25)
df <- data.frame(needs, ease, emotion, cx_date, cx_group)
cxi_trend(df, cx_date, cx_group)
```

---

nps\_calc

*Tidy Calculation of Net Promoter Score*


---

**Description**

Simplifies the calculation of Net Promoter Score (NPS) from raw survey data within the tidyverse framework.

**Usage**

```
nps_calc(survey_data, ...)
```

**Arguments**

survey_data	Raw survey data. Needs to be one row per survey with the nps question in a numeric column called nps_question responses having column names of needs, ease and emotion
...	optional columns by which to group the nps calculation. There is no limit to the number of grouping variables chosen. Too many will likely result in nps calculations that are too fragmented / based on very small survey counts.

**Details**

Net Promoter Score was originally developed by Fred Reichheld and now is owned by Bain Company and Satmetrix Systems. According to Wikipedia it "is a management tool that can be used to gauge the loyalty of a firm's customer relationships."

**Value**

Data frame with nps and survey count for each combination of the grouping variables

**Examples**

```
nps_question <- sample(10, 100, replace = TRUE)
nps_date <- rep(seq.Date(from = as.Date("2019-01-01"), to = as.Date("2019-01-10"), by = "day"), 10)
nps_group <- rep(c("a", "b", "c", "d"), 25)
df <- data.frame(nps_question, nps_date, nps_group)
nps_calc(df, nps_group)
```

nps\_trend

*Tidy Calculation of Net Promoter Score trends by group***Description**

Simplifies the calculation of Net Promoter Score (NPS) trends over time from raw survey data within the tidyverse framework.

**Usage**

```
nps_trend(survey_data, trend_var, ..., min_surveys = 1, avg_surveys = 0)
```

**Arguments**

survey_data	Raw survey data. Needs to be one row per survey with the nps question in a numeric column called nps_question
trend_var	Column that represents an element of time, eg week number, date, month & year
...	optional columns by which to group the nps calculation. There is no limit to the number of grouping variables chosen. Too many will likely result in nps calculations that are too fragmented / based on very small survey counts.
min_surveys	Minimum surveys found in every period for each group to be included
avg_surveys	Average surveys found in every period for each group to be included

**Details**

Net Promoter Score was originally developed by Fred Reichheld and now is owned by Bain Company and Satmetrix Systems. According to Wikipedia it "is a management tool that can be used to gauge the loyalty of a firm's customer relationships."

The trend version of the function allows you to specify one column as a date over which to trend the data. This allows quick filtering to eliminate groupings that fail to meet user-specified thresholds for average or minimum survey counts per time period.

The resulting data set is set up for creating faceted line plots using ggplot2.

**Value**

Data frame with nps and survey count for each combination of the grouping variables over the time variable.

**Examples**

```
nps_question <- sample(10, 100, replace = TRUE)
nps_date <- rep(seq.Date(from = as.Date("2019-01-01"), to = as.Date("2019-01-10"), by = "day"), 10)
nps_group <- rep(c("a", "b", "c", "d"), 25)
df <- data.frame(nps_question, nps_date, nps_group)
nps_trend(df, nps_date, nps_group)
```

# Index

`cxi_calc`, 2  
`cxi_trend`, 3

`nps_calc`, 4  
`nps_trend`, 5