

Package ‘pricesensitivitymeter’

October 14, 2022

Type Package

Title Van Westendorp Price Sensitivity Meter Analysis

Version 1.2.2

Maintainer Max Alletsee <max.alletsee@gmail.com>

Description An implementation of the van Westendorp Price Sensitivity Meter in R, which is a survey-based approach to analyze consumer price preferences and sensitivity (van Westendorp 1976, isbn:9789283100386).

License MIT + file LICENSE

Encoding UTF-8

BugReports <https://github.com/max-alletsee/pricesensitivitymeter/issues>

URL <https://max-alletsee.github.io/pricesensitivitymeter/>,
<https://github.com/max-alletsee/pricesensitivitymeter>

Depends R (>= 3.0.0), methods

Imports survey (>= 3.23-1), ggplot2, rlang

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Max Alletsee [aut, cre]

Repository CRAN

Date/Publication 2021-10-19 19:20:02 UTC

R topics documented:

pricesensitivitymeter	2
psm-class	2
psm_analysis	4
psm_analysis_weighted	9
psm_plot	13
summary.psm	15

Index

17

pricesensitivitymeter *Consumer Price Preferences and Price Sensitivity Analysis*

Description

pricesensitivitymeter is an implementation of the van Westendorp Price Sensitivity Meter method to analyze consumer price preferences and price sensitivity in R. Besides the estimation of optimal price points and price ranges, it can also model the optimal price in terms of reach or revenue (based on the so-called Newton Miller Smith extension).

To read the documentation for the function's syntax, see [psm_analysis](#) and [psm_analysis_weighted](#) (for weighted data).

Author(s)

Max Alletsee <max.alletsee@gmail.com>

References

Van Westendorp, P (1976) "NSS-Price Sensitivity Meter (PSM) – A new approach to study consumer perception of price" *Proceedings of the 29th ESOMAR Congress*, 139–167. Online available at <https://www.researchworld.com/a-new-approach-to-study-consumer-perception-of-price/>.

Newton, D, Miller, J, Smith, P, (1993) "A market acceptance extension to traditional price sensitivity measurement" *Proceedings of the American Marketing Association Advanced Research Techniques Forum*.

See Also

Useful links:

- <https://github.com/max-alletsee/pricesensitivitymeter>
- Report bugs at <https://github.com/max-alletsee/pricesensitivitymeter/issues>

psm-class

Class "psm"

Description

Class "psm" is a class for outputs of Price Sensitivity Meter analyses as performed by the [psm_analysis](#) function.

The main purpose is to create a custom summary function for objects of class "psm".

Objects from the Class

Objects are usually created as a result of a call of the [psm_analysis](#) function.

Slots

- data_input:** Object of class "data.frame". Contains the data that was used as an input for the analysis.
- validated:** Object of class "logical". Indicates whether the "validate" option of the `psm_analysis` function has been used (to exclude cases with intransitive price preferences).
- invalid_cases:** Object of class "numeric". Number of cases with intransitive price preferences.
- total_sample:** Object of class "numeric". Total sample size of the input sample *before* assessing the transitivity of individual price preferences.
- data_vanwestendorp:** Object of class "data.frame". Output data of the Price Sensitivity Meter analysis. Contains the cumulative distribution functions for the four price assessments (too cheap, cheap, expensive, too expensive) for all prices as well as the inversed distributions "not cheap" and "not expensive" (that are required for the acceptable price range).
- pricerange_lower:** Object of class "numeric". Lower limit of the acceptable price range as defined by the Price Sensitivity Meter, also known as **point of marginal cheapness**: Intersection of the "too cheap" and the "not cheap" curves.
- pricerange_upper:** Object of class "numeric". Upper limit of the acceptable price range as defined by the Price Sensitivity Meter, also known as **point of marginal expensiveness**: Intersection of the "too expensive" and the "not expensive" curves.
- idp:** Object of class "numeric". **Indifference Price Point** as defined by the Price Sensitivity Meter: Intersection of the "cheap" and the "expensive" curves.
- opp:** Object of class "numeric". **Optimal Price Point** as defined by the Price Sensitivity Meter: Intersection of the "too cheap" and the "too expensive" curves.
- weighted:** Object of class "logical". TRUE if the function has used weighted data to calculate the price points; FALSE if unweighted data has been used.
- survey_design:** Object of class "survey.design2". If weighted data has been used, the survey design object from the survey package is returned here. Please refer to the documentation in the survey package for more details.
- NMS:** Object of class "logical". Indicates whether the additional analyses of the Newton Miller Smith Extension were performed.
- data_nms:** Object of class "data.frame". Output of the Newton Miller Smith extension: calibrated mean purchase probabilities for each price point.
- pi_scale:** Object of class "data.frame". Shows the values of the purchase intent variable and the corresponding calibrated purchase probabilities as defined in the function input for the Newton Miller Smith extension.
- price_optimal_trial:** Object of class "numeric". Output of the Newton Miller Smith extension: Estimate for the price with the highest trial rate.
- price_optimal_revenue:** Object of class "numeric". Output of the Newton Miller Smith extension: Estimate for the price with the highest revenue (based on the trial rate).

Methods

[summary.psm](#)

References

Van Westendorp, P (1976) "NSS-Price Sensitivity Meter (PSM) – A new approach to study consumer perception of price" *Proceedings of the 29th ESOMAR Congress*, 139–167. Online available at <https://www.researchworld.com/a-new-approach-to-study-consumer-perception-of-price/>.

Newton, D, Miller, J, Smith, P, (1993) "A market acceptance extension to traditional price sensitivity measurement" *Proceedings of the American Marketing Association Advanced Research Techniques Forum*.

See Also

To understand the main function that creates an object of class "psm", see [psm_analysis](#) or [psm_analysis_weighted](#).

To understand how the summaries of objects of class "psm" look like, see [summary.psm](#).

For a documentation of objects of class "survey.design2", see the documentation of the survey package.

Examples

```
showClass("psm")
```

psm_analysis

Van Westendorp Price Sensitivity Meter Analysis (PSM)

Description

`psm_analysis()` performs an analysis of consumer price preferences and price sensitivity known as **van Westendorp Price Sensitivity Meter (PSM)**. It takes respondent's price preferences (from survey data) as an input and estimates acceptable price ranges and price points. For a description of the method see the *Details* section.

Usage

```
psm_analysis(  
  toocheap, cheap, expensive, tooexpensive,  
  data = NA,  
  validate = TRUE,  
  interpolate = FALSE,  
  interpolation_steps = 0.01,  
  intersection_method = "min",  
  pi_cheap = NA, pi_expensive = NA,  
  pi_scale = 5:1,  
  pi_calibrated = c(0.7, 0.5, 0.3, 0.1, 0))
```

Arguments

toocheap, cheap, expensive, tooexpensive	<p>If a data.frame/matrix/tibble is provided in the data argument: names of the variables in the data.frame/matrix that contain the survey data on the respondents' "too cheap", "cheap", "expensive" and "too expensive" price preferences.</p> <p>If no data.frame/matrix/tibble is provided in the data argument: numeric vectors that directly include this information. If numeric vectors are provided, it is assumed that they are sorted by respondent ID (the preferences for respondent n are stored at the n-th position in all vectors).</p> <p>If the toocheap price was not assessed, a variable/vector of NAs can be used instead. This variable/vector needs to have the same length as the other survey information. If toocheap is NA for all cases, it is possible to calculate the Point of Marginal Expensiveness and the Indifference Price Point, but it is impossible to calculate the Point of Marginal Cheapness and the Optimal Price Point.</p>
data	<p>data.frame, matrix or tibble that contains the function's input data. data input is not mandatory: Instead of using a data.frame/matrix/tibble as an input, it is also possible to provide the data directly as vectors in the "too cheap", "cheap", "expensive" and "too expensive" arguments.</p>
validate	<p>logical. should only respondents with consistent price preferences (too cheap < cheap < expensive < too expensive) be considered in the analysis?</p>
interpolate	<p>logical. should interpolation of the price curves be applied between the actual prices given by the respondents? If interpolation is enabled, the output appears less bumpy in regions with sparse price information. If the sample size is sufficiently large, interpolation should not be necessary.</p>
interpolation_steps	<p>numeric. if interpolate is TRUE: the size of the interpolation steps. Set by default to 0.01, which should be appropriate for most goods in a price range of 0-50 USD/Euro.</p>
intersection_method	<p>"min" (default), "max", "mean" or "median". defines the method how to determine the price points (range, indifference price, optimal price) if there are multiple possible intersections of the price curves. "min" uses the lowest possible prices, "max" uses the highest possible prices, "mean" calculates the mean among all intersections and "median" uses the median of all possible intersections</p>
pi_cheap, pi_expensive	<p>Only required for the Newton Miller Smith extension. If data argument is provided: names of the variables in the data.frame/matrix/tibble that contain the survey data on the respondents' purchase intent at their individual cheap/expensive price.</p>
pi_scale	<p>Only required for the Newton Miller Smith extension. Scale of the purchase intent variables pi_cheap and pi_expensive. By default assuming a five-point scale with 5 indicating the highest purchase intent.</p>
pi_calibrated	<p>Only required for the Newton Miller Smith extension. Calibrated purchase probabilities that are assumed for each value of the purchase intent scale. Must be the same order as the pi_scale variable so that the first value of pi_calibrated</p>

corresponds to the first value in the `pi_scale` variable. Default values are taken from the Sawtooth Software PSM implementation in Excel: 70% for the best value of the purchase intent scale, 50% for the second best value, 30% for the third best value (middle of the scale), 10% for the fourth best value and 0% for the worst value.

Details

The Price Sensitivity Meter method for the analysis of consumer price preferences was proposed by the Dutch economist Peter van Westendorp in 1976 at the ESOMAR conference. It is a survey-based approach that has become one of the standard price acceptance measurement techniques in the market research industry and is still widely used for during early-stage product development.

Price acceptance and price sensitivity are measured in van Westendorp's approach by four open-ended survey questions:

- At which price on this scale are you beginning to experience ... (test-product) as cheap?
- At which price on this scale are you beginning to experience ... (test-product) as expensive?
- At which price on this scale you are beginning to experience ... (test-product) as too expensive – so that you would never consider buying it yourself?
- At which price on this scale you are beginning to experience ... (test-product) as too cheap – so that you say "at this price the quality cannot be good"?

Respondents with inconsistent price preferences (e.g. "cheap" price larger than "expensive" price) are usually removed from the data set. This function has built-in checks to detect invalid preference structures and removes those respondents from the analysis by default.

To analyze price preferences and price sensitivity, the method uses cumulative distribution functions for each of the aforementioned price steps (e.g. "how many respondents think that a price of x or more is expensive?"). By convention, the distributions for the "too cheap" and the "cheap" price are inverted. This leads to the interpretation "how many respondents think that a price of *up to* x is (too) cheap?".

The interpretation is built on the analysis of the intersections of the four cumulative distribution functions for the different prices (usually via graphical inspection). The original paper describes the four intersections as follows:

- **Point of Marginal Cheapness (MGP):** Below this price point, there are more respondents that consider the price as "too cheap" than respondents who consider it as "not cheap" (intersection of "too cheap" and "not cheap"). This is interpreted as the lower limit of the range of acceptable prices.
- **Point of Marginal Expensiveness (MEP):** Above this price point, there are more respondent that consider the price as "too expensive" than there are respondents who consider it as "not expensive" (intersection of "not expensive" and "too expensive"). This is interpreted as the upper limit of the range of acceptable prices.
- **Indifference Price Point (IDP):** The same number of respondents perceives the price as "cheap" and "expensive" (intersection of "cheap" and "expensive"). In van Westendorp's interpretation, this is either the median price paid in the market or the price of an important market-leader.

- **Optimal Price Point (OPP):** The same number of respondents perceives the product as "too cheap" and "too expensive" (intersection of "too cheap" and "too expensive"). van Westendorp argues that this is the value for which the respondents' resistance against the price is particularly low.

Besides those four intersections, van Westendorp's article advises to analyze the cumulative distribution functions for steep areas which indicate price steps.

To analyze trial rates and estimate revenue forecasts, Newton/Miller/Smith have extended van Westendorp's original model by adding two purchase intent questions that are asked for the respondent's "cheap" and "expensive" price. The purchase probability at the respondent's "too cheap" and "too expensive" price are defined as 0. The main logic is that the "too expensive" price point is prohibitively expensive for the respondent and a price at the "too cheap" price level raises doubts about the product quality.

By combining the standard van Westendorp questions with those two additional purchase intent questions, it becomes possible to summarize the purchase probabilities across respondents (using linear interpolation for the purchase probabilities between each respondent's cornerstone prices). The maximum of this curve is then defined as the price point with the highest expected trial rate. Moreover, by multiplying the trial rate with the price, it also becomes possible to estimate a price with the highest expected revenue.

It has to be noted that the van Westendorp Price Sensitivity Meter is useful in some cases, but does not answer every pricing-related question. It may be a good tool to assess very broadly if the consumers' price perceptions exceed the actual production costs. For more complex analyses (e.g. defining specific prices for different products to avoid cannibalization and drive at the same time incremental growth), other methodological approaches are needed.

Value

The function output consists of the following elements:

`data_input`: `data.frame` object. Contains the data that was used as an input for the analysis.

`validated`: logical object. Indicates whether the "validate" option has been used (to exclude cases with intransitive price preferences).

`invalid_cases`: numeric object. Number of cases with intransitive price preferences.

`total_sample`: "numeric" object. Total sample size of the input sample *before* assessing the transitivity of individual price preferences.

`data_vanwestendorp`: `data.frame` object. Output data of the Price Sensitivity Meter analysis. Contains the cumulative distribution functions for the four price assessments (too cheap, cheap, expensive, too expensive) for all prices.

`pricerange_lower`: numeric object. Lower limit of the acceptable price range as defined by the Price Sensitivity Meter, also known as **point of marginal cheapness**: Intersection of the "too cheap" and the "not cheap" curves.

`pricerange_upper`: numeric object. Upper limit of the acceptable price range as defined by the Price Sensitivity Meter, also known as **point of marginal expensiveness**: Intersection of the "too expensive" and the "not expensive" curves.

idp:	numeric object. Indifference Price Point as defined by the Price Sensitivity Meter: Intersection of the "cheap" and the "expensive" curves.
opp:	numeric object. Optimal Price Point as defined by the Price Sensitivity Meter: Intersection of the "too cheap" and the "too expensive" curves.
NMS:	logical object. Indicates whether the additional analyses of the Newton Miller Smith Extension were performed.
weighted:	logical object. Indicates if weighted data was used in the analysis. Outputs from psm_analysis() always have the value FALSE. When data is weighted, use the function psm_analysis_weighted .
data_nms:	data.frame object. Output of the Newton Miller Smith extension: calibrated mean purchase probabilities for each price point.
pi_scale:	data.frame object. Shows the values of the purchase intent variable and the corresponding calibrated purchase probabilities as defined in the function input for the Newton Miller Smith extension.
price_optimal_trial:	numeric object. Output of the Newton Miller Smith extension: Estimate for the price with the highest trial rate.
price_optimal_revenue:	numeric object. Output of the Newton Miller Smith extension: Estimate for the price with the highest revenue (based on the trial rate).

References

- Van Westendorp, P (1976) "NSS-Price Sensitivity Meter (PSM) – A new approach to study consumer perception of price" *Proceedings of the ESOMAR 29th Congress*, 139–167. Online available at <https://www.researchworld.com/a-new-approach-to-study-consumer-perception-of-price/>.
- Newton, D, Miller, J, Smith, P, (1993) "A market acceptance extension to traditional price sensitivity measurement" *Proceedings of the American Marketing Association Advanced Research Techniques Forum*.
- Sawtooth Software (2016) "Templates for van Westendorp PSM for Lighthouse Studio and Excel". Online available at <https://sawtoothsoftware.com/resources/software-downloads/tools/van-westendorp-price-sensitivity-meter>

See Also

The function `psm_analysis_weighted()` performs the same analyses for weighted data.

Examples

```
set.seed(42)

# standard van Westendorp Price Sensitivity Meter Analysis
# input directly via vectors

tch <- round(rnorm(n = 250, mean = 5, sd = 0.5), digits = 2)
ch <- round(rnorm(n = 250, mean = 8.5, sd = 0.5), digits = 2)
```



```

ex <- round(rnorm(n = 250, mean = 13, sd = 0.75), digits = 2)
tex <- round(rnorm(n = 250, mean = 17, sd = 1), digits = 2)

output_psm_demo1 <- psm_analysis(toocheap = tch,
  cheap = ch,
  expensive = ex,
  tooexpensive = tex)

# additional analysis with Newton Miller Smith Extension
# input via data.frame

pint_ch <- sample(x = c(1:5), size = length(tex),
  replace = TRUE, prob = c(0.1, 0.1, 0.2, 0.3, 0.3))

pint_ex <- sample(x = c(1:5), size = length(tex),
  replace = TRUE, prob = c(0.3, 0.3, 0.2, 0.1, 0.1))

data_psm_demo <- data.frame(tch, ch, ex, tex, pint_ch, pint_ex)

output_psm_demo2 <- psm_analysis(toocheap = "tch",
  cheap = "ch",
  expensive = "ex",
  tooexpensive = "tex",
  pi_cheap = "pint_ch",
  pi_expensive = "pint_ex",
  data = data_psm_demo)

summary(output_psm_demo2)

```

psm_analysis_weighted *Weighted van Westendorp Price Sensitivity Meter Analysis (PSM)*

Description

psm_analysis_weighted() performs a **weighted** analysis of consumer price preferences and price sensitivity known as **van Westendorp Price Sensitivity Meter (PSM)**. The function requires a sample design from the **survey** package as the main input. Custom weights or sample designs from other packages are not supported.

To run a PSM analysis **without** weighting, use the function [psm_analysis](#).

Usage

```

psm_analysis_weighted(
  toocheap, cheap, expensive, tooexpensive,
  design,
  validate = TRUE,
  interpolate = FALSE,
  interpolation_steps = 0.01,
  intersection_method = "min",

```

```
pi_cheap = NA, pi_expensive = NA,
pi_scale = 5:1,
pi_calibrated = c(0.7, 0.5, 0.3, 0.1, 0))
```

Arguments

- tocheap, cheap, expensive, tooexpensive**
Names of the variables in the data.frame/matrix that contain the survey data on the respondents' "too cheap", "cheap", "expensive" and "too expensive" price preferences.
If the tocheap price was not assessed, a variable of NAs can be used instead. If tocheap is NA for all cases, it is possible to calculate the Point of Marginal Expensiveness and the Indifference Price Point, but it is impossible to calculate the Point of Marginal Cheapness and the Optimal Price Point.
- design**
A survey design which has been created by the function `svydesign()` from the **survey** package. The data that is used as an input of `svydesign()` must include all the variable names for tocheap, cheap, expensive and tooexpensive variables specified above.
- validate**
logical. should only respondents with consistent price preferences (too cheap < cheap < expensive < too expensive) be considered in the analysis?
- interpolate**
logical. should interpolation of the price curves be applied between the actual prices given by the respondents? If interpolation is enabled, the output appears less bumpy in regions with sparse price information. If the sample size is sufficiently large, interpolation should not be necessary.
- interpolation_steps**
numeric. if interpolate is TRUE: the size of the interpolation steps. Set by default to 0.01, which should be appropriate for most goods in a price range of 0-50 USD/Euro.
- intersection_method**
"min" (default), "max", "mean" or "median". defines the method how to determine the price points (range, indifference price, optimal price) if there are multiple possible intersections of the price curves. "min" uses the lowest possible prices, "max" uses the highest possible prices, "mean" calculates the mean among all intersections and "median" uses the median of all possible intersections
- pi_cheap, pi_expensive**
Only required for the Newton Miller Smith extension. Names of the variables in the data that contain the survey data on the respondents' purchase intent at their individual cheap/expensive price.
- pi_scale**
Only required for the Newton Miller Smith extension. Scale of the purchase intent variables pi_cheap and pi_expensive. By default assuming a five-point scale with 5 indicating the highest purchase intent.
- pi_calibrated**
Only required for the Newton Miller Smith extension. Calibrated purchase probabilities that are assumed for each value of the purchase intent scale. Must be the same order as the pi_scale variable so that the first value of pi_calibrated corresponds to the first value in the pi_scale variable. Default values are taken from the Sawtooth Software PSM implementation in Excel: 70% for the best

value of the purchase intent scale, 50% for the second best value, 30% for the third best value (middle of the scale), 10% for the fourth best value and 0% for the worst value.

Details

The main logic of the Price Sensitivity Meter Analysis is explained in the documentation of the [psm_analysis](#) function. The `psm_analysis_weighted` performs the same analysis, but weights the survey data according to a known population.

Value

The function output consists of the following elements:

<code>data_input:</code>	<code>data.frame</code> object. Contains the data that was used as an input for the analysis.
<code>validated:</code>	logical object. Indicates whether the "validate" option has been used (to exclude cases with intransitive price preferences).
<code>invalid_cases:</code>	numeric object. Number of cases with intransitive price preferences.
<code>total_sample:</code>	"numeric" object. Total sample size of the input sample <i>before</i> assessing the transitivity of individual price preferences.
<code>data_vanwestendorp:</code>	<code>data.frame</code> object. Output data of the Price Sensitivity Meter analysis. Contains the weighted cumulative distribution functions for the four price assessments (too cheap, cheap, expensive, too expensive) for all prices.
<code>pricerange_lower:</code>	numeric object. Lower limit of the acceptable price range as defined by the Price Sensitivity Meter, also known as point of marginal cheapness : Intersection of the "too cheap" and the "expensive" curves.
<code>pricerange_upper:</code>	numeric object. Upper limit of the acceptable price range as defined by the Price Sensitivity Meter, also known as point of marginal expensiveness : Intersection of the "too expensive" and the "cheap" curves.
<code>idp:</code>	numeric object. Indifference Price Point as defined by the Price Sensitivity Meter: Intersection of the "cheap" and the "expensive" curves.
<code>opp:</code>	numeric object. Optimal Price Point as defined by the Price Sensitivity Meter: Intersection of the "too cheap" and the "too expensive" curves.
<code>weighted:</code>	logical object. Indicating if weighted data was used in the analysis. Outputs from <code>psm_analysis_weighted()</code> always have the value TRUE. When data is unweighted, use the function psm_analysis .
<code>survey_design:</code>	<code>survey.design2</code> object. Returning the full survey design as specified with the svydesign function from the survey package.
<code>NMS:</code>	logical object. Indicates whether the additional analyses of the Newton Miller Smith Extension were performed.

References

Van Westendorp, P (1976) "NSS-Price Sensitivity Meter (PSM) – A new approach to study consumer perception of price" *Proceedings of the ESOMAR 29th Congress*, 139–167. Online available at <https://www.researchworld.com/a-new-approach-to-study-consumer-perception-of-price/>.

Newton, D, Miller, J, Smith, P, (1993) "A market acceptance extension to traditional price sensitivity measurement" *Proceedings of the American Marketing Association Advanced Research Techniques Forum*.

Sawtooth Software (2016) "Templates for van Westendorp PSM for Lighthouse Studio and Excel". Online available at <https://sawtoothsoftware.com/resources/software-downloads/tools/van-westendorp-price-sensitivity-meter>

Examples

```
# assuming a skewed sample with only 1/3 women and 2/3 men

input_data <- data.frame(tch = round(rnorm(n = 250, mean = 8, sd = 1.5), digits = 2),
  ch = round(rnorm(n = 250, mean = 12, sd = 2), digits = 2),
  ex = round(rnorm(n = 250, mean = 13, sd = 1), digits = 2),
  tex = round(rnorm(n = 250, mean = 15, sd = 1), digits = 2),
  gender = sample(x = c("male", "female"),
    size = 250,
    replace = TRUE,
    prob = c(2/3, 1/3)))

# ... and in which women have on average 1.5x the price acceptance of men
input_data$tch[input_data$gender == "female"] <- input_data$tch[input_data$gender == "female"] * 1.5
input_data$ch[input_data$gender == "female"] <- input_data$ch[input_data$gender == "female"] * 1.5
input_data$ex[input_data$gender == "female"] <- input_data$ex[input_data$gender == "female"] * 1.5
input_data$tex[input_data$gender == "female"] <- input_data$tex[input_data$gender == "female"] * 1.5

# creating a sample design object using the survey package
# ... assuming that gender is balanced equally in the population of 10000

input_data$gender_pop <- 5000

input_design <- survey::svydesign(ids = ~ 1, # no clusters
  probs = NULL, # hence no cluster sampling probabilities,
  strata = input_data$gender, # stratified by gender
  fpc = input_data$gender_pop, # strata size in the population
  data = input_data)
# data object used as input: no need to specify single variables

output_weighted_psm <- psm_analysis_weighted(toocheap = "tch",
  cheap = "ch",
  expensive = "ex",
  tooexpensive = "tex",
  design = input_design)

summary(output_weighted_psm)
```

psm_plot

*Plot of the van Westendorp Price Sensitivity Meter Analysis (PSM)***Description**

psm_plot() uses ggplot() to show the standard van Westendorp Price Sensitivity Meter plot that allows to see the acceptance for each price on each of the four variables.

It takes the object created by psm_analysis() or psm_analysis_weighted() as an input.

Usage

```
psm_plot(psm_result,
        shade_pricerange = TRUE,
        line_toocheap = TRUE,
        line_tooexpensive = TRUE,
        line_notcheap = TRUE,
        line_notexpensive = TRUE,
        point_idp = TRUE,
        point_color_idp = "#009E73",
        label_idp = TRUE,
        point_opp = TRUE,
        point_color_opp = "#009E73",
        label_opp = TRUE,
        pricerange_color = "grey50",
        pricerange_alpha = 0.3,
        line_color = c("too cheap" = "#009E73",
                      "not cheap" = "#009E73",
                      "not expensive" = "#D55E00",
                      "too expensive" = "#D55E00"),
        line_type = c("too cheap" = "dotted",
                     "not cheap" = "solid",
                     "not expensive" = "solid",
                     "too expensive" = "dotted"))
```

Arguments

psm_result	Result of a Price Sensitivity Meter analysis, created by running psm_analysis() or psm_analysis_weighted(). (Object of class "psm")
shade_pricerange	logical value. Determines if the acceptable price range is shown as a shaded area or not.
line_toocheap	logical value. Determines if the line for the "too cheap" price curve is shown or not.
line_tooexpensive	logical value. Determines if the line for the "too expensive" price curve is shown or not.

<code>line_notcheap</code>	logical value. Determines if the line for the "not cheap" price curve is shown or not.
<code>line_notexpensive</code>	logical value. Determines if the line for the "not expensive" price curve is shown or not.
<code>point_idp</code>	logical value. Determines if the Indifference Price Point is shown or not.
<code>point_color_idp</code>	character vector, specifying the color of the Optimal Price Point. Can be a hex color (e.g. "#7f7f7f") or one of R's built-in colors (e.g. "grey50").
<code>label_idp</code>	logical value. Determines if the label for the Indifference Price Point is shown or not.
<code>point_opp</code>	logical value. Determines if the Optimal Price Point is shown or not.
<code>point_color_opp</code>	character vector, specifying the color of the Optimal Price Point. Can be a hex color (e.g. "#7f7f7f") or one of R's built-in colors (e.g. "grey50").
<code>label_opp</code>	logical value. Determines if the label for the Optimal Price Point is shown or not.
<code>pricerange_color</code>	character, specifying the background color for the accepted price range. Can be a hex color (e.g. "#7f7f7f") or one of R's built-in colors (e.g. "grey50"). You can see all of R's built-in colors with the function <code>colors()</code> . Is only applied if <code>shade_pricerange = TRUE</code>
<code>pricerange_alpha</code>	numeric between 0 and 1, specifying the alpha transparency for the shaded area of the the accepted price range. Is only applied if <code>shade_pricerange = TRUE</code>
<code>line_color</code>	character vector, specifying the line color for each of the price curves shown. Color definitions must match the lines you have defined via <code>line_toocheap</code> , <code>line_tooexpensive</code> , <code>line_notcheap</code> and <code>line_notexpensive</code> . Can be a hex color (e.g. "#7f7f7f") or one of R's built-in colors (e.g. "grey50").
<code>line_type</code>	vector, specifying the line type for each of the price curves shown. Definitions must match the lines you have defined via <code>line_toocheap</code> , <code>line_tooexpensive</code> , <code>line_notcheap</code> and <code>line_notexpensive</code> . Values must match <code>ggplot2</code> 's expectations for line types: An integer (0:8), a name (blank, solid, dashed, dotted, dotdash, longdash, twodash), or a string with an even number (up to eight) of hexadecimal digits which give the lengths in consecutive positions in the string.

Value

The function output is a `ggplot2` object.

References

Van Westendorp, P (1976) "NSS-Price Sensitivity Meter (PSM) – A new approach to study consumer perception of price" *Proceedings of the ESOMAR 29th Congress*, 139–167. Online available at <https://www.researchworld.com/a-new-approach-to-study-consumer-perception-of-price/>.

See Also

The vignette "Visualizing PSM Results" shows a similar way and more custom way to plot the data.

Examples

```
# set up example data and run psm_analysis()

tch <- round(rnorm(n = 250, mean = 5, sd = 0.5), digits = 2)
ch <- round(rnorm(n = 250, mean = 8.5, sd = 0.5), digits = 2)
ex <- round(rnorm(n = 250, mean = 13, sd = 0.75), digits = 2)
tex <- round(rnorm(n = 250, mean = 17, sd = 1), digits = 2)

output_psm_demo <- psm_analysis(toocheap = tch,
  cheap = ch,
  expensive = ex,
  tooexpensive = tex)

# create the plot (note that ggplot's convention
# is to *not* show it by default)
## Not run: psm_result_plot <- psm_plot(output_psm_demo)

# to show the plot, call the object (and maybe
# additional ggplot functions if you like)
psm_result_plot + ggplot2::theme_minimal()
## End(Not run)
```

summary.psm

Summarizing Price Sensitivity Meter Analyses

Description

summary method for class "psm".

Usage

```
## S3 method for class 'psm'
summary(object, ...)
```

Arguments

object an object of class "psm", usually a result of a call to [psm_analysis](#) or [psm_analysis_weighted](#).
 ... further arguments from other methods - will be ignored in the summary

References

Van Westendorp, P (1976) "NSS-Price Sensitivity Meter (PSM) – A new approach to study consumer perception of price" *Proceedings of the 29th ESOMAR Congress*, 139–167. Online available at <https://www.researchworld.com/a-new-approach-to-study-consumer-perception-of-price/>.

Newton, D, Miller, J, Smith, P, (1993) "A market acceptance extension to traditional price sensitivity measurement" *Proceedings of the American Marketing Association Advanced Research Techniques Forum*.

See Also

A comprehensive description of all elements of an object of class "psm" can be found in [psm_analysis](#) and [psm_analysis_weighted](#).

The generic [summary](#) function.

Index

* classes

- psm-class, [2](#)
- pricesensitivitymeter, [2](#)
- pricesensitivitymeter-package
 - (pricesensitivitymeter), [2](#)
- psm-class, [2](#)
- psm.class (psm-class), [2](#)
- psm_analysis, [2](#), [4](#), [4](#), [9](#), [11](#), [15](#), [16](#)
- psm_analysis_weighted, [2](#), [4](#), [8](#), [9](#), [15](#), [16](#)
- psm_plot, [13](#)
- summary, [16](#)
- summary.psm, [3](#), [4](#), [15](#)
- svydesign, [10](#), [11](#)