

# Package ‘pollster’

August 25, 2020

**Type** Package

**Title** Calculate Crosstab and Topline Tables of Weighted Survey Data

**Version** 0.1.3

**Author** John D. Johnson [aut, cre]

**Maintainer** John D. Johnson <john.d.johnson@marquette.edu>

**Description** Calculate common types of tables for weighted survey data. Options include topline and (2-way and 3-way) crosstab tables of categorical or ordinal data as well as summary tables of weighted numeric variables. Optionally, include the margin of error at selected confidence intervals including the design effect. The design effect is calculated as described by Kish (1965) <doi:10.1002/bimj.19680100122> beginning on page 257. Output takes the form of tibbles (simple data frames). This package conveniently handles labelled data, such as that commonly used by 'Stata' and 'SPSS.' Complex survey design is not supported at this time.

**Depends** R (>= 2.10)

**Imports** dplyr (>= 0.8.0), stringr (>= 1.0.0), tidyr (>= 1.1.0), labelled (>= 2.0.0), forcats, rlang (>= 0.4.5)

**Suggests** ggplot2 (>= 3.3.0), knitr, rmarkdown

**License** CC0

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.0

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-08-25 09:50:02 UTC

## R topics documented:

crosstab . . . . .	2
crosstab_3way . . . . .	3
deff_calc . . . . .	4
illinois . . . . .	5
moedeff_calc . . . . .	6
moe_crosstab . . . . .	6
moe_crosstab_3way . . . . .	8
moe_topline . . . . .	9
moe_wave_crosstab . . . . .	10
moe_wave_crosstab_3way . . . . .	11
summary_table . . . . .	12
topline . . . . .	13
wtd_mean . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

crosstab	<i>weighted crosstabs</i>
----------	---------------------------

---

### Description

crosstab returns a tibble containing a weighted crosstab of two variables

### Usage

```
crosstab(
  df,
  x,
  y,
  weight,
  remove = "",
  n = TRUE,
  pct_type = "row",
  format = "wide",
  unwt_n = FALSE
)
```

### Arguments

df	The data source
x	The independent variable
y	The dependent variable
weight	The weighting variable

remove	An optional character vector of values to remove from final table (e.g. "refused"). This will not affect any calculations made. The vector is not case-sensitive.
n	logical, if TRUE numeric totals are included. They are included in a separate column for row and cell percentages, but in a separate row for wide format column percentages.
pct_type	Controls the kind of percentage values returned. One of "row," "cell," or "column."
format	one of "long" or "wide"
unwt_n	logical, if TRUE a column "unweighted_n" is included containing the unweighted frequency count. It is not available when pct_type is "column"

### Details

Options include row, column, or cell percentages. The tibble can be in long or wide format.

### Value

a tibble

### Examples

```
crosstab(df = illinois, x = voter, y = raceethnic, weight = weight)
crosstab(df = illinois, x = voter, y = raceethnic, weight = weight, n = FALSE)
```

---

crosstab_3way	<i>weighted 3-way crosstabs</i>
---------------	---------------------------------

---

### Description

crosstab\_3way returns a tibble containing a weighted crosstab of two variables by a third variable

### Usage

```
crosstab_3way(
  df,
  x,
  y,
  z,
  weight,
  remove = c(""),
  n = TRUE,
  pct_type = "row",
  format = "wide",
  unwt_n = FALSE
)
```

**Arguments**

df	The data source
x	The independent variable
y	The dependent variable
z	The second control variable
weight	The weighting variable
remove	An optional character vector of values to remove from final table (e.g. "refused"). This will not affect any calculations made. The vector is not case-sensitive.
n	logical, if TRUE numeric totals are included.
pct_type	Controls the kind of percentage values returned. One of "row" or "cell."
format	one of "long" or "wide"
unwt_n	logical, if TRUE a column is added containing unweighted frequency counts

**Details**

Options include row or cell percentages. The tibble can be in long or wide format. These tables are ideal for use with small multiples created with `ggplot2::facet_wrap`.

**Value**

a tibble

**Examples**

```
crosstab_3way(df = illinois, x = sex, y = educ6, z = maritalstatus, weight = weight)
crosstab_3way(df = illinois, x = sex, y = educ6, z = maritalstatus, weight = weight,
format = "wide")
```

---

deff\_calc

*Calculate the design effect of a sample*


---

**Description**

deff\_calc returns a single number

**Usage**

```
deff_calc(w)
```

**Arguments**

w a vector of weights

**Details**

This function returns the design effect of a given sample using the formula  $\text{length}(w) * \text{sum}(w^2) / (\text{sum}(w))^2$ . It is designed for use in the moe family of functions.

**Value**

A number

**Examples**

```
deff_calc(illinois$weight)
```

---

illinois	<i>Illinois respondents to the Voting and Registration Supplement for the Current Population Survey</i>
----------	---

---

**Description**

A dataset containing the responses of 36,207 Illinois respondents to the Current Population Survey's biennial Voting and Registration Supplement for the Current Population Survey, 1996-2018.

**Usage**

```
illinois
```

**Format**

A data frame with 36207 rows and 9 variables:

**year** year of survey

**fips** the state fips code

**sex** sex of the respondent, labelled value

**educ6** highest level of education for respondent, labelled values

**raceethnic** one of white, black, Hispanic, or other, labelled values

**maritalstatus** one of Married, Widowed/divorced/Sep, or Never Married, labelled values

**rv** indicates if the respondent is registered to vote, labelled values

**voter** indicates if the respondent voted, labelled values

**age** the age of the respondent, numeric values

**weight** the number of people each respondent is calculated to represent

**Source**

<https://www.census.gov/topics/public-sector/voting.html>

---

moedeff_calc	<i>Calculate the margin of error (including design effect) of a sample</i>
--------------	--

---

### Description

moedeff\_calc returns a single number. It is designed for use in the moe family of functions.

### Usage

```
moedeff_calc(pct, deff, n, zscore = 1.96)
```

### Arguments

pct	a proportion
deff	a design effect
n	the sample size
zscore	defaults to 1.96, consistent with a 95% confidence interval.

### Details

This function returns the margin of error including design effect of a given sample of weighted data using the formula  $\text{sqrt}(\text{deff}) * \text{zscore} * \text{sqrt}(\text{pct} * (1 - \text{pct}) / (n - 1)) * 100$

### Value

A percentage

### Examples

```
moedeff_calc(pct = 0.515, deff = 1.6, n = 214)
```

---

moe_crosstab	<i>weighted crosstabs with margin of error</i>
--------------	--

---

### Description

moe\_crosstab returns a tibble containing a weighted crosstab of two variables with margin of error

**Usage**

```
moe_crosstab(
  df,
  x,
  y,
  weight,
  remove = c(""),
  n = TRUE,
  pct_type = "row",
  format = "long",
  zscore = 1.96,
  unwt_n = FALSE
)
```

**Arguments**

df	The data source
x	The independent variable
y	The dependent variable
weight	The weighting variable, defaults to <code>zwave_weight</code>
remove	An optional character vector of values to remove from final table (e.g. "refused"). This will not affect any calculations made. The vector is not case-sensitive.
n	logical, if TRUE numeric totals are included.
pct_type	Controls the kind of percentage values returned. One of "row" or "cell." Column percents are not supported.
format	one of "long" or "wide"
zscore	defaults to 1.96, consistent with a 95% confidence interval
unwt_n	logical, if TRUE it adds a column with unweighted frequency values

**Details**

Options include row or cell percentages. The tibble can be in long or wide format. The margin of error includes the design effect of the weights.

**Value**

a tibble

**Examples**

```
moe_crosstab(df = illinois, x = voter, y = raceethnic, weight = weight)
moe_crosstab(df = illinois, x = voter, y = raceethnic, weight = weight, n = FALSE)
```

---

moe\_crosstab\_3way      *weighted 3-way crosstabs with margin of error*

---

### Description

moe\_crosstab\_3way returns a tibble containing a weighted crosstab of two variables by a third variable with margin of error

### Usage

```
moe_crosstab_3way(
  df,
  x,
  y,
  z,
  weight,
  remove = c(""),
  n = TRUE,
  pct_type = "row",
  format = "long",
  zscore = 1.96,
  unwt_n = FALSE
)
```

### Arguments

df	The data source
x	The independent variable
y	The dependent variable
z	The second control variable
weight	The weighting variable
remove	An optional character vector of values to remove from final table (e.g. "refused"). This will not affect any calculations made. The vector is not case-sensitive.
n	logical, if TRUE numeric totals are included.
pct_type	Controls the kind of percentage values returned. One of "row" or "cell."
format	one of "long" or "wide"
zscore	defaults to 1.96, consistent with a 95% confidence interval
unwt_n	logical, if TRUE it adds a column with unweighted frequency values

### Details

Options include row or cell percentages. The tibble can be in long or wide format. These tables are ideal for use with small multiples created with `ggplot2::facet_wrap`.

**Value**

a tibble

**Examples**

```
moe_crosstab_3way(df = illinois, x = sex, y = educ6, z = maritalstatus, weight = weight)
moe_crosstab_3way(df = illinois, x = sex, y = educ6, z = maritalstatus, weight = weight,
format = "wide")
```

---

moe_topline	<i>weighted topline with margin of error</i>
-------------	--

---

**Description**

moe\_topline returns a tibble containing a weighted topline of one variable with margin of error

**Usage**

```
moe_topline(
  df,
  variable,
  weight,
  remove = c(""),
  n = TRUE,
  pct = TRUE,
  valid_pct = TRUE,
  cum_pct = TRUE,
  zscore = 1.96
)
```

**Arguments**

df	The data source
variable	the variable name
weight	The weighting variable, defaults to zwave_weight
remove	An optional character vector of values to remove from final table (e.g. "refused"). This will not affect any calculations made. The vector is not case-sensitive.
n	logical, if TRUE a frequency column is included percentages, but in a separate row for column percentages.
pct	logical, if TRUE a column of percents is included
valid_pct	logical, if TRUE a column of valid percents is included
cum_pct	logical, if TRUE a column of cumulative percents is included
zscore	defaults to 1.96, consistent with a 95% confidence interval

**Details**

By default the table includes a column for frequency count, percent, valid percent, and cumulative percent.

**Value**

a tibble

**Examples**

```
moe_topline(df = illinois, variable = educ6, weight = weight)
moe_topline(df = illinois, variable = educ6, weight = weight, remove = c("LT HS"))
```

---

moe_wave_crosstab	<i>weighted crosstabs with margin of error, where the x-variable identifies different survey waves</i>
-------------------	--

---

**Description**

moe\_wave\_crosstab returns a tibble containing a weighted crosstab of two variables with margin of error. Use this function when the x-variable indicates different survey waves for which weights were calculated independently.

**Usage**

```
moe_wave_crosstab(
  df,
  x,
  y,
  weight,
  remove = c(""),
  n = TRUE,
  pct_type = "row",
  format = "long",
  zscore = 1.96,
  unwt_n = FALSE
)
```

**Arguments**

df	The data source
x	The independent variable, which uniquely identifies survey waves
y	The dependent variable
weight	The weighting variable, defaults to zwave_weight
remove	An optional character vector of values to remove from final table (e.g. "refused"). This will not affect any calculations made. The vector is not case-sensitive.

n	logical, if TRUE numeric totals are included.
pct_type	Controls the kind of percentage values returned. One of "row" or "cell." Column percents are not supported.
format	one of "long" or "wide"
zscore	defaults to 1.96, consistent with a 95% confidence interval
unwt_n	logical, if TRUE it adds a column with unweighted frequency values

### Details

Options include row or cell percentages. The tibble can be in long or wide format. The margin of error includes the design effect of the weights, calculated separately for each survey wave.

### Value

a tibble

### Examples

```
moe_wave_crosstab(df = illinois, x = year, y = maritalstatus, weight = weight)
moe_wave_crosstab(df = illinois, x = year, y = maritalstatus, weight = weight, format = "wide")
```

---

moe\_wave\_crosstab\_3way

*weighted 3-way crosstabs with margin of error, where the z-variable identifies different survey waves*

---

### Description

moe\_wave\_crosstab\_3way returns a tibble containing a weighted crosstab of two variables by a third variable with margin of error. Use this function when the z-variable indicates different survey waves for which weights were calculated independently.

### Usage

```
moe_wave_crosstab_3way(
  df,
  x,
  y,
  z,
  weight,
  remove = c(""),
  n = TRUE,
  pct_type = "row",
  format = "long",
  zscore = 1.96,
  unwt_n = FALSE
)
```

**Arguments**

df	The data source
x	The independent variable
y	The dependent variable
z	The second control variable, uniquely identifies survey waves
weight	The weighting variable
remove	An optional character vector of values to remove from final table (e.g. "re-fused"). This will not affect any calculations made. The vector is not case-sensitive.
n	logical, if TRUE numeric totals are included.
pct_type	Controls the kind of percentage values returned. One of "row" or "cell."
format	one of "long" or "wide"
zscore	defaults to 1.96, consistent with a 95% confidence interval
unwt_n	logical, if TRUE it adds a column with unweighted frequency values

**Details**

Options include row or cell percentages. The tibble can be in long or wide format. These tables are ideal for use with small multiples created with `ggplot2::facet_wrap`.

**Value**

a tibble

**Examples**

```
moe_crosstab_3way(df = illinois, x = sex, y = educ6, z = year, weight = weight)
moe_crosstab_3way(df = illinois, x = sex, y = educ6, z = year, weight = weight, format = "wide")
```

---

summary_table	<i>weighted summary table</i>
---------------	-------------------------------

---

**Description**

`summary_table` returns a tibble containing a weighted summary table of a single variable.

**Usage**

```
summary_table(df, variable, weight, name_style = "clean")
```

**Arguments**

<code>df</code>	The data source
<code>variable</code>	the variable to summarize, it should be numeric
<code>weight</code>	The weighting variable
<code>name_style</code>	the style of the column names—one of "clean" or "pretty." Clean names are all lower case and words are separated by an underscore. Pretty names begin with a capital letter and words are separated by a space.

**Details**

The resulting tibble includes columns for the variable name, unweighted observations, weighted observations, weighted mean, minimum value, maximum value, unweighted missing values, and weighted missing values

**Value**

a tibble

**Examples**

```
summary_table(illinois, age, weight)
summary_table(illinois, age, weight, name_style = "pretty")
```

---

topline	<i>weighted topline</i>
---------	-------------------------

---

**Description**

topline returns a tibble containing a weighted topline of one variable

**Usage**

```
topline(
  df,
  variable,
  weight,
  remove = c(""),
  n = TRUE,
  pct = TRUE,
  valid_pct = TRUE,
  cum_pct = TRUE
)
```

**Arguments**

df	The data source
variable	the variable name
weight	The weighting variable, defaults to zwave_weight
remove	An optional character vector of values to remove from final table (e.g. "refused"). This will not affect any calculations made. The vector is not case-sensitive.
n	logical, if TRUE a frequency column is included percentages, but in a separate row for column percentages.
pct	logical, if TRUE a column of percents is included
valid_pct	logical, if TRUE a column of valid percents is included
cum_pct	logical, if TRUE a column of cumulative percents is included

**Details**

By default the table includes a column for frequency count, percent, valid percent, and cumulative percent.

**Value**

a tibble

**Examples**

```
topline(illinois, sex, weight)
topline(illinois, sex, weight, pct = FALSE)
```

---

wtd_mean	<i>weighted mean</i>
----------	----------------------

---

**Description**

wtd\_mean returns the weighted mean of a variable. It's a tidy-compatible wrapper around stats::weighted.mean().

**Usage**

```
wtd_mean(df, variable, weight)
```

**Arguments**

df	The data source
variable	the variable, it should be numeric
weight	The weighting variable

**Value**

a numeric value

**Examples**

```
wtd_mean(illinois, age, weight)
```

```
library(dplyr)  
illinois %>% wtd_mean(age, weight)
```

# Index

## \* datasets

illinois, [5](#)

crosstab, [2](#)

crosstab\_3way, [3](#)

deff\_calc, [4](#)

illinois, [5](#)

moe\_crosstab, [6](#)

moe\_crosstab\_3way, [8](#)

moe\_topline, [9](#)

moe\_wave\_crosstab, [10](#)

moe\_wave\_crosstab\_3way, [11](#)

moedeff\_calc, [6](#)

summary\_table, [12](#)

topline, [13](#)

wtd\_mean, [14](#)