Package 'manylabRs'

September 14, 2017

Type Package

Version 0.1.0

Title Analysis scripts and data from the Many Labs projects

Date 2017-09-12

Maintainer Fred Hasselman <me@fredhasselman.com>

Description Code and data from the ManyLabs projects. ML1: Investigating Variation in Replicability (https://osf.io/wx7ck/); ML2: Investigating Variation in Replicability Across Sample and Setting (https://osf.io/8cd4r/).

License GPL-3

Encoding UTF-8

LazyData true

Imports tidyverse

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

---

any2any *any2any*

Description

Converts most common test statistics into most common (signed) effect sizes.

Usage

```
any2any(testInfo, df1 = NULL, df2 = NULL, N = NULL, n1 = NULL,
  n2 = NULL, esType = NA, var.lor = NA, CIcalc = TRUE, CL = 0.95,
  rID = 0, q = 1, alternative = "two", keepDirection = TRUE,
  keepSign = TRUE, keepSignNames = c("r", "l.r", "u.r", "fisher.z", "l.z",
  "u.z"))
```

Arguments

df1	Degrees of freedom
df2	NULL or degrees of freedom of the denominator for the f-distribution.
N	Number of data points used in calculation of test-statistic.
n1	Number of data points in sample 1.
n2	Number of data points in sample 2.
esType	Type of test statistic. One of: "t", "lm.t", "f", "lm.f", "r", "X2", "Z", "lm.Z"
CIcalc	If TRUE (default) the Confidence Interval for the test statistic in x will be calculated using the "Confidence limits for noncentral parameters" functions in package (e.g., for type - "t": conf.limits.nct).
CL	Confidence Limit (default: .95).
rID	Correlation among predictor values in a linear model.
q	Number of predictors in the model.
alternative	Alternative hypothesis (defult = "two").
keepSign	Return effect size with sign of test statistic? (default = TRUE).
keepSignNames	Which effect sizes should keep the sign if keepSign = TRUE? Default is to keep the sign for: "r","l.r","u.r","fisher.z","l.z","u.z".
st	Value(s) of a test statistic.

Details

The procedure to calculate a variety of effect sizes is as follows:

- If CIcalc == FALSE, package::compute.es will be used to convert the test statistic to a large number of effect size estimates. The confidence intervals around the effect size estimates will be based meta-analytic estimates of effect size variance (e.g., for type - "t": tes).

- If `CIcalc == TRUE`, `package::MBESS` will be used to calculate the confidence interval for the test statistic based on its noncentral distribution (e.g., for type - "t": conf.limits.nct). Subsequently the test statistic, as well as its lower and upper confidence limit will each be passed to `compute.es` seperately.
- If `keepSign == TRUE` the sign of the test statistic will be copied to all the effect sizes in `keepSignNames`.

Value

The effect sizes calculated by `compute.es` corresponding to the test statistic(s), with either meta-analytic, or, exact CI.

Note

The prefix "lm" is currently disregarded, but will be implemented in future versions to indicate the test statistic is in fact a fixed factor in a linear model.

Author(s)

F Hasselman (inspired by RP:P function `any2r` by CHJ Hartgerink)

`cor.test.fisherZ` *cor.test.fisherZ*

Description

cor.test.fisherZ

Usage

```
## S3 method for class 'fisherZ'
cor.test(r1 = NULL, r2 = NULL, n1 = NULL, n2 = NULL,
  p = TRUE, Cohens.q = TRUE, conf.level = 0.95,
  alternative = "two.sided", null.value = 0, cor.type = "pearson")
```

Arguments

`r1`	First correlation
`r2`	Second correlation
`n1`	First sample size
`n2`	Second sample size
`p`	Compute p-value? (default = TRUE)
`Cohens.q`	Compute effect size Cohen's q (default = TRUE)
`alternative`	One of "greater", "less", "two.sided" (default)
`alpha`	Alpha evel for significance test

disp *disp*

Description

Displays easy-to-spot text in the Console.

Usage

```
disp(message = "Hello world!", header = "disp", footer = TRUE)
```

Arguments

message	A message to be displayed in the Console.
header	Print a header of '~' symbols (=TRUE), or '~' symbols with few words of text (=character vector)
footer	Print a footer '~' symbols.

fill_viol *fill_viol*

Description

This is adapted from: http://stackoverflow.com/questions/22278951/combining-violin-plot-with-box-plo

Usage

```
fill_viol(gr.df, gr, qtile, probs)
```

Arguments

gr.df	Internal.
gr	Internal.
qtile	Internal.
probs	Internal.

Details

Function to create geom_ploygon calls for vioQtile

Value

A list for vioQtile

See Also

vioQtile

`get.analyses` *get.analyses*

Description

get.analyses

Usage

```
get.analyses(studies = NA, analysis.type = NA, Nmin.raw = 30,
  Nmin.cond = 15, subset = c("all", "WEIRD", "NON-WEIRD")[1],
  rootdir = "~/Dropbox/Manylabs2/TestOutput", indir = list(RAW.DATA =
  "RAW.DATA.PRIVATE", MASTERKEY = "", SOURCEINFO = ""), outdir = list(ROBJECTS
  = "ROBJECTS", RESULTS.RDS = "RESULTS.RDS"))
```

Arguments

`studies`	Numeric vector with unique study IDs listed in the 'masteRkey' table (default = all IDs).
`Nmin.raw`	Minimum raw sample size allowed to be included in the analyses.
`Nmin.cond`	Minimum sample size per condition allowed to be included in the analyses.
`tp`	An optional number indicating Global (1), Primary (2, default) or Secondary (3) analyses.

Details

Run analyses for (selected) ML2 studies.

Value

A list object with analysis results.

See Also

Other "get." functions: `get.GoogleSheet`, `get.OSFfile`, `get.Order`, `get.zavCode`

`get.cases` *get.cases*

Description

get.cases

Usage

```
get.cases(rule, study.vars, study.vars.labels, stat.params)
```

Arguments

rule	Internal
study.vars	Internal
study.vars.labels	Internal
stat.params	Internal

---

get.chain *get.chain*

Description

get.chain

Usage

```
get.chain(inf)
```

Arguments

inf	Internal

---

get.CSVdata *get.CSVdata*

Description

get.CSVdata

Usage

```
get.CSVdata(path, fID, finishedOnly = TRUE)
```

Arguments

path	Path to the data.
finishedOnly	Only import cases with value of variable `Finished = 1` (default).
files	A list of `.csv` / `.xlsx` files containing raw ML2 data.

get.GoogleSheet *get.GoogleSheet*

Description

get.GoogleSheet

Usage

```
get.GoogleSheet(url = NULL, data = c("ML1data", "ML2masteRkey",
  "ML2data")[2], dfCln = FALSE, Sep = ".")
```

Arguments

url	Hyperlink to the GoogleSheet, ending in command ".../export?format=csv".
data	If no URL is provided, which dataset? (default = "ML2masteRkey").
dfCln	Should the variable names be cleaned (replace spaces and punctuation by a period "."). Default is FALSE.
Sep	Symbol to use when changing column names (default: ".").

Value

A list object with fields:

- Returned if dataSet = TRUE (default):
  - df: A data table generated by tbl_df from package dplyr.
  - info: Information about the downloaded file including a time stamp, the URL and original row and column names.
- Returned if dataSet = FALSE:
  - FilePath: The local path to the downloaded file.

See Also

Other "get." functions: get.OSFfile, get.Order, get.analyses, get.zavCode

get.info *get.info*

Description

get.info

Usage

```
get.info(keytable, cols, subset)
```

Arguments

keytable	Internal
cols	Internal

get.ncpCI *get.ncpCI*

Description

get.ncpCI

Usage

```
get.ncpCI(x, df1, df2, N, esType, CL = 0.95, keepSign = TRUE,
  keepDirection = TRUE, alternative = "two.sided")
```

Arguments

x	A noncentrality parameter.
df1	Degrees of freeddom.
df2	NULL or degrees of freedom of the denominator for the f-distribution.
N	Sample size
esType	Type of test statistic. One of: "t", "t.r", lm.t", "f", "lm.f", "r", "X2", "Z", "lm.Z"
CL	Confidence Limit (default: .95).
keepSign	Return effect size with sign of test statistic? (default = TRUE).
keepDirection	Use the information in `alternative` to decide on one-sided vs. two-sided confidence intervals. Default is `TRUE`. If `FALSE`, two-sided CIs will be calclulated irrespective of the direction of the `alternative`.
alternative	Alternative hypothesis (defult = "two").

get.Order *get.Order*

Description

get.Order

Usage

```
get.Order(df, S1 = TRUE)
```

Arguments

df	A ManyLabs2 data frame.
S1	Are the data from slate1 (default) or slate2?

Value

A list object with fields:

- `df`: ManyLabs2 data frame in which the Qualtrics study order has been added to each case.
- `Problems`: Cases for which the study order information could not be retrieved.

Author(s)

Fred Hasselman

See Also

Other "get." functions: `get.GoogleSheet`, `get.OSFfile`, `get.analyses`, `get.zavCode`

`get.OSFfile` *get.OSFfile*

Description

get.OSFfile

Usage

```
get.OSFfile(code, dir = tempdir(), scanMethod, downloadMethod = c("httr",
  "downloader", "curl"), dataSet = TRUE, dfCln = FALSE)
```

Arguments

`code`	Either a full url ("https://osf.io/XXXXX/"), or just the OSF code.
`dir`	Output location (default is `tempdir()`).
`scanMethod`	Either `readLines` or `RCurl`. Leave missing to choose automatically.
`downloadMethod`	One of `httr` (default), `downloader` or `curl`.
`dataSet`	Is the file data set which can be imported using `import` from package `rio`?
`dfCln`	Should the variable names be cleaned (replace spaces and punctuation by a period "."). Default is `FALSE`.

Details

Function to download a file hosted on OSF. Modified from code originally written by Sacha Epskamp.

Value

A list object with fields:

- Returned if `dataSet = TRUE` (default):
  - `df`: A data table generated by `tbl_df` from package `dplyr`.
  - `info`: Information about the downloaded file including a time stamp, the URL and original row and column names.
- Returned if `dataSet = FALSE`:
  - `FilePath`: The local path to the downloaded file.

Author(s)

Fred Hasselman, based on code by Sasha Epskamp

See Also

Other "get." functions: `get.GoogleSheet`, `get.Order`, `get.analyses`, `get.zavCode`

get.plotly *get.plotly*

Description

get.plotly

Usage

```
get.plotly(data, analysis_url)
```

Arguments

data	Dataframe with ML2 testresutls and ESCI output.

get.sourceData *get.sourceData*

Description

get.sourceData

Usage

```
get.sourceData(ML2.id, ML2.df, ML2.in)
```

Arguments

ML2.id	Internal
ML2.df	Internal
ML2.in	Inernal

Value

A list with fields study.vars (data organised according to the masteRkey spreadsheet), study.vars/labels, N, and RawDataFilter(raw data, unfiltered).

get.zavCode *get.zavCode*

Description

get.zavCode

Usage

```
get.zavCode(df = NULL, lookup = NULL)
```

Arguments

df	Dataset: ML2.S2
lookup	The lookup table

Value

The code for each sentence.

See Also

Other "get." functions: get.GoogleSheet, get.OSFfile, get.Order, get.analyses

gg.plotHolder *gg.plotHolder*

Description

gg.plotHolder

Usage

```
gg.plotHolder(useArial = F, afmPATH = "~/Dropbox")
```

Arguments

useArial	Use the Arial font (requires .afm font files in the afmPath)
afmPATH	Path to Arial .afm font files.

Value

A blank ggplot2 object that can be used in concordance with grid.arrange.

Examples

```
# Create a plot with marginal distributions.
library(ggplot2)
library(scales)

df <- data.frame(x = rnorm(n = 100), y = rnorm(n = 100),
                 group = factor(sample(x=c(0,1),
                 size = 100, replace = TRUE))
                 )

scatterP <- ggplot(df, aes(x = x, y =y, colour = group)) +
            geom_point() +
            gg.theme()

xDense <- ggplot(df, aes(x = x, fill = group)) +
          geom_density(aes(y= ..count..),trim=FALSE, alpha=.5) +
          gg.theme("noax") +
          theme(legend.position = "none")

yDense <- ggplot(df, aes(x = y, fill = group)) +
          geom_density(aes(y= ..count..),trim=FALSE, alpha=.5) +
          coord_flip() +
          gg.theme("noax") +
          theme(legend.position = "none")

library(gridExtra)

grid.arrange(xDense,
             gg.plotHolder(),
             scatterP,
             yDense,
             ncol=2, nrow=2,
             widths=c(4, 1.4), heights=c(1.4, 4)
             )
```

---

gg.theme *gg.theme*

Description

gg.theme

Usage

```
gg.theme(type = c("clean", "noax"), useArial = F, afmPATH = "~/Dropbox")
```

Arguments

type	One of "clean", or "noax"
useArial	Use the Arial font (requires .afm font files in the afmPath)
afmPATH	Path to Arial .afm font files.

Details

Will generate a "clean" ggplot theme, or a theme without any axes ("noax").

Some scientific journals explicitly request the Arial font should be used in figures. This can be achieved by using .afm font format (see, e.g. http://www.pure-mac.com/font.html).

Value

A theme for ggplot2.

Examples

```
library(ggplot2)
g <- ggplot(data.frame(x = rnorm(n = 100), y = rnorm(n = 100)), aes(x = x, y = y)) + geom_point()
g + gg.theme()
g + gg.theme("noax")
```

---

in.IT *Initialise It*

Description

Load and/or install R packages

Usage

```
in.IT(need = NULL, inT = TRUE)
```

Arguments

need	A vector of package names to be loaded. The wrapper functions have a predefinded need list and can be used as shortcuts (see details).
inT	Logical. If TRUE (default), packages in need wil be installed if they are not available on the system.

Details

in.IT will check if the Packages in the list argument need are installed on the system and load them. If inT=TRUE (default), it will first install the packages if they are not present and then proceed to load them.

Author(s)

Fred Hasselman

See Also

Other initialise packages: un.IT

Examples

```
in.IT(c("reshape2", "plyr", "dplyr"))
```

renderHTMLresults *renderHTMLresults*

Description

renderHTMLresults

Usage

```
renderHTMLresults(pageID)
```

Arguments

pageID

scaleR *scaleR*

Description

Rescale a vector to a user defined range defined by user.

Usage

```
scaleR(x, mn = min(x, na.rm = T), mx = max(x, na.rm = T), lo = 0,
  hi = 1)
```

Arguments

x	Input vector or data frame.
mn	Minimum value of original, defaults to `min(x, na.rm = TRUE)`.
mx	Maximum value of original, defaults to `max(x, na.rm = TRUE)`.
lo	Maximum value to rescale to, defaults to `1`.
hi	Minimum value to rescale to, defaults to `0`.

Details

Three uses:

1. scaleR(x) - Scale x to data range: min(x.out)==0; max(x.out)==1
2. scaleR(x,mn,mx) - Scale x to arg. range: min(x.out)==mn==0; max(x.out)==mx==1
3. scaleR(x,mn,mx,lo,hi) - Scale x to arg. range: min(x.out)==mn==lo; max(x.out)==mx==hi

Author(s)

Fred Hasselman

Examples

```
# Works on numeric objects
somenumbers <- cbind(c(-5,100,sqrt(2)),c(exp(1),0,-pi))

scaleR(somenumbers)
scaleR(somenumbers,mn=-100)
# Values < mn will return < lo (default=0)
# Values > mx will return > hi (default=1)
scaleR(somenumbers,mn=-1,mx=99)

scaleR(somenumbers,lo=-1,hi=1)
scaleR(somenumbers,mn=-10,mx=101,lo=-1,hi=4)
```

---

testScript *testScript*

Description

FOR TESTING PURPOSES

Usage

```
testScript(studies, tp, saveCSVfile = NA, saveRDSfile = NA,
  subset = c("all", "WEIRD", "NONWEIRD")[1],
  dir.out = "~/Dropbox/Manylabs2/TestOutput")
```

Arguments

studies	Unique analysis number(s) from the matsterkey sheet.
tp	Analysis type (1 = 'study.global.include', 2 = 'study.primary.include', 3 = 'study.secondary.include').
saveRDSfile	Save an RDS file of the output.

---

try.CATCH *try.CATCH both warnings (with value) and errors*

Description

In longer simulations, aka computer experiments, you may want to 1) catch all errors and warnings (and continue) 2) store the error or warning messages

Here's a solution (see R-help mailing list, Dec 9, 2010):

Catch *and* save both errors and warnings, and in the case of a warning, also keep the computed result.

Usage

```
try.CATCH(expr)
```

Arguments

expr	an R expression to evaluate

Value

a list with 'value' and 'warning', where value' may be an error caught.

Author(s)

Martin Maechler; Copyright (C) 2010-2012 The R Core Team

`un.IT` *Un-initialise It*

Description

Unload and/or uninstall R packages.

Usage

```
un.IT(loose, unT = FALSE)
```

Arguments

`loose`	A vector of package names to be unloaded.
`unT`	Logical. If `TRUE`, packages in `loose` wil be un-installed if they are available on the system.

Details

`un.IT`will check if the Packages in the list argument `loose` are installed on the system and unload them. If `unT=TRUE` it will first unload the packages if they are loaded, and then proceed to uninstall them.

Author(s)

Fred Hasselman

See Also

Other initialise packages: `in.IT`

Examples

```
## Not run: un.IT(loose = c("reshape2", "plyr", "dplyr"), unT = FALSE)
```

varfun.Alter.1 *varfun.Alter.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#3_alter

Usage

```
varfun.Alter.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

Syllogisms to include for each sample INCLUSION PERCENTAGE BASED ON FLUENT / DISFLUENT SEPERATELY: 1 5 6 BOTH: 1 5 6

varfun.Alter.2 *varfun.Alter.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#3_alter

Usage

```
varfun.Alter.2(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

Syllogisms to include for each sample INCLUSION PERCENTAGE BASED ON FLUENT / DISFLUENT SEPERATELY: 1 5 6 BOTH: 1 5 6

varfun.Alter.3 *varfun.Alter.3*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#3_alter

Usage

```
varfun.Alter.3(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

Syllogisms to include for each sample INCLUSION PERCENTAGE BASED ON FLUENT / DISFLUENT SEPERATELY: 1 5 6 BOTH: 1 5 6 @examples

varfun.Alter.4 *varfun.Alter.4*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#3_alter

Usage

```
varfun.Alter.4(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

Syllogisms to include for each sample INCLUSION PERCENTAGE BASED ON FLUENT / DISFLUENT SEPERATELY: 1 5 6 BOTH: 1 5 6

`varfun.Anderson.1` *varfun.Anderson.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#12_anderson

Usage

```
varfun.Anderson.1(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

and1.3=Satisfaction With Life Scale (SWLS, 5 items, Low SocioMetricStatus condition), higher numbers=higher satisfaction; and1.4=Positive And Negative Affect Scale (PANAS, Low SocioMetricStatus condition). Positive items are 1,4,5,8,9,12,14,17,18,19. Negative items: 2,3,6,7,10,11,13,15,16,20; Alert: recode responses to negative items before averaging. and2.3=Satisfaction With Life Scale (SWLS, 5 items, High SocioMetricStatus condition), higher numbers=higher satisfaction; and2.4=Positive And Negative

Affect Scale (PANAS, High SocioMetricStatus condition). Positive items are 1,4,5,8,9,12,14,17,18,19. Negative items: 2,3,6,7,10,11,13,15,16,20; Alert: recode responses to negative items before averaging.

list(Low=c("and1.3_1", "and1.3_2", "and1.3_3", "and1.3_4", "and1.3_5", "and1.4_1", "and1.4_2", "and1.4_3", "and1.4_4", "and1.4_5", "and1.4_6", "and1.4_7", "and1.4_8", "and1.4_9", "and1.4_10", "and1.4_11", "and1.4_12", "and1.4_13", "and1.4_14", "and1.4_15", "and1.4_16", "and1.4_17", "and1.4_18", "and1.4_19", "and1.4_20"), High=c("and2.3_1", "and2.3_2", "and2.3_3", "and2.3_4", "and2.3_5", "and2.4_1", "and2.4_2", "and2.4_3", "and2.4_4", "and2.4_5", "and2.4_6", "and2.4_7", "and2.4_8", "and2.4_9", "and2.4_10", "and2.4_11", "and2.4_12", "and2.4_13", "and2.4_14", "and2.4_15", "and2.4_16", "and2.4_17", "and2.4_18", "and2.4_19", "and2.4_20"))

`varfun.Bauer.1` *varfun.Bauer.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#6_bauer

Usage

```
varfun.Bauer.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

---

varfun.Critcher.1 *varfun.Critcher.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#9_critcher

Usage

```
varfun.Critcher.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

crit1.1= crit2.1=

df.P97 <- dplyr::select(tbl_df(ML2.df),which(colnames(ML2.df) df.P97 <- slice(df.P97, which((ML2.id[[1]][,1]==T)&(

df.P17 <- dplyr::select(tbl_df(ML2.df),which(colnames(ML2.df) df.P17 <- slice(df.P17, which((ML2.id[[1]][,2]==T)&(

id.P97 <- ML2.df[ML2.id[[1]][,1]&ML2.id[[2]][,1],ML2.in$study.vars$Condition[1]] id.P17 <- ML2.df[ML2.id[[1]][,2

---

varfun.Gati.1 *varfun.Gati.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#28_gati

Usage

```
varfun.Gati.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

References

Tversky, A., & Gati, I. (1978). Studies of similarity. **Cognition and categorization**, 1, 79-98.

`varfun.Gati.2` *varfun.Gati.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#28_gati

Usage

```
varfun.Gati.2(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

References

Tversky, A., \& Gati, I. (1978). Studies of similarity. **Cognition and categorization**, 1, 79-98.

`varfun.Gati.3` *varfun.Gati.3*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#28_gati

Usage

```
varfun.Gati.3(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

References

Tversky, A., & Gati, I. (1978). Studies of similarity. **Cognition and categorization**, 1, 79-98.

`varfun.Gati.4` *varfun.Gati.4*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#28_gati

Usage

```
varfun.Gati.4(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

Note

This analysis tests moderating effect of presenting Norenzayan first or after.

References

Tversky, A., & Gati, I. (1978). Studies of similarity. **Cognition and categorization**, 1, 79-98.

`varfun.Gati.5` *varfun.Gati.5*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#28_gati

Usage

```
varfun.Gati.5(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

References

Tversky, A., & Gati, I. (1978). Studies of similarity. **Cognition and categorization**, 1, 79-98.

`varfun.Giessner.1` *varfun.Giessner.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#17_giessner

Usage

```
varfun.Giessner.1(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

geis.1.1=long line condition; geis.2.1=short line condition; geis.dv_1=dominant; geis.dv_2=strong; geis.dv_3=self-confident; geis.dv_4=control; geis.dv_5=status; For all dvs, higher numbers=higher power.

`varfun.Graham.1` *varfun.Graham.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#4_graham

Usage

```
varfun.Graham.1(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

varfun.Graham.2 *varfun.Graham.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#4_graham

Usage

```
varfun.Graham.2(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

varfun.Gray.1 *varfun.Gray.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#22_gray

Usage

```
varfun.Gray.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

Adult harms baby scenario; gray1.2=responsibility (adult); gray1.4=pain (baby). Baby harms adult scenario; gray2.2=responsibility (baby); gray2.4=pain (adult)

References

Gray, K., & Wegner, D. M. (2009). Moral typecasting: divergent perceptions of moral agents and moral patients. Journal of Personality and Social Psychology, 96, 505.

varfun.Gray.2 *varfun.Gray.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#22_gray

Usage

```
varfun.Gray.2(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

Baby harms adult scenario; gray1.3=intentionality (adult); gray1.4=pain (baby). Adult harms baby scenario; gray2.3=intentionality (baby); gray2.4=pain (adult)

References

Gray, K., & Wegner, D. M. (2009). Moral typecasting: divergent perceptions of moral agents and moral patients. Journal of Personality and Social Psychology, 96, 505.

varfun.Hauser.1 *varfun.Hauser.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#11_hauser

Usage

```
varfun.Hauser.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

haus1.1t = timing (side effect scenario); haus2.1t = timing (greater good scenario); haus1.2=previous experience (drop if 1 (yes)); haus2.2=previous experience (drop if 1 (yes)); haus1.1=morally permissible (side effect scenario; Yes=1); haus2.1=morally permissible (greater good scenario; Yes=1).

`varfun.Hauser.2` *varfun.Hauser.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#17_hauser

Usage

```
varfun.Hauser.2(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

hauser3.1=morality judgment (greater good condition); hauser4.1=morality judgment (foreseen side-effect condition; for both, yes=1, no=2.) haus3.2 and haus4.2=previous experience (yes=1, no=2); haus3.1t_3=timing (greater good); haus4.1t_3=timing (side effect).

References

Hauser, M., Cushman, F., Young, L., Kang-Xing Jin, R., & Mikhail, J. (2007). A dissociation between moral judgments and justifications. Mind & Language, 22, 1-21.

`varfun.Hsee.1` *varfun.Hsee.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#21_hsee

Usage

```
varfun.Hsee.1(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

hsee1.1=generosity ($90 scarf condition); hsee2.1=generosity ($110 coat condition); for both, higher numbers=higher generosity

References

Hsee, C. K. (1998). Less is better: When low-value options are valued more highly than high-value options. Journal of Behavioral Decision Making, 11, 107-121.

`varfun.Huang.1` *Huang.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#1_huang

Usage

```
varfun.Huang.1(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis A `list` object containing fields **High**, **Low** and **N**

Variables

huan1.1_Y1: Y position of the mouse (High SES condition). huan2.1_Y1: Y position of the mouse (Low SES). huan1.1_R0 and huan2.1_R0 indicate for each condition whether a click was inside the map (1) or outside (0).

For each condition a participant must have clicked inside the map (=1) to be included in the analysis.

varfun.Inbar.1 *varfun.Inbar.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#8_inbar

Usage

```
varfun.Inbar.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

disg1.11,disg1.12,disg2.10,disg2.12,disg2.13;

responses on the DS-R are scored as follows: True 1, False 0; Not disgusting 0, Slightly disgusting 0.5, Very disgusting 1

varfun.Inbar.2 *varfun.Inbar.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#8_inbar

Usage

```
varfun.Inbar.2(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

varfun.Kay.1 *varfun.Kay.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#2_kay

Usage

```
varfun.Kay.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

varfun.Knobe.1 *varfun.Knobe.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#27_knobe

Usage

```
varfun.Knobe.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

knob1.3=intentionality (help condition); knob2.3=intentionality (harm condition); for both, higher numbers=higher intentionality

References

Knobe, J. (2003). Intentional action and side effects in ordinary language. Analysis, 63, 190-193.

`varfun.Knobe.2` *varfun.Knobe.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#27_knobe

Usage

```
varfun.Knobe.2(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

knob1.4=intentionality (praise condition); knob2.4=intentionality (blame condition); for both, higher numbers=higher intentionality

References

Knobe, J. (2003). Intentional action and side effects in ordinary language. Analysis, 63, 190-193.

`varfun.Miyamoto.1` *varfun.Miyamoto.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#7_miyamoto

Usage

```
varfun.Miyamoto.1(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Details

Analysis plan: An ANCOVA will compare the mean estimates of the author's true attitude across the two conditions, covarying for perceived constraint.

Value

Dataset ready for analysis

Variables

miya1.5=true attitude (pro-death condition; higher values=higher support for death penalty); miya1.7=perceived constraint (pro-death condition; higher values=higher freedom);

miya2.5=true attitude (against death penalty condition; higher values=higher support for death penalty); miya2.7=perceived constraint (against death condition; higher values= higher freedom).

`varfun.Miyamoto.2` *varfun.Miyamoto.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#7_miyamoto

Usage

```
varfun.Miyamoto.2(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

miya1.5=true attitude (pro-death condition; higher values=higher support for death penalty); miya1.7=perceived constraint (pro-death condition; higher values=higher freedom);

miya2.5=true attitude (against death penalty condition; higher values=higher support for death penalty); miya2.7=perceived constraint (against death condition; higher values= higher freedom).

`varfun.Norenzayan.1` *varfun.Norenzayan.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#20_norenzayan

Usage

```
varfun.Norenzayan.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

nore1.1 TO nore1.20 provide choices ("belong to" condition)

nore2.1 to nore2.20 provide choices ("similar to" condition)

References

Norenzayan, A., Smith, E. E., Kim, B. J., & Nisbett, R. E. (2002). Cultural preferences for formal versus intuitive reasoning. Cognitive Science, 26, 653-684.

---

varfun.Norenzayan.2 *varfun.Norenzayan.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#20_norenzayan

Usage

```
varfun.Norenzayan.2(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Note

This analysis tests moderating effect of presenting Gati first or after.

References

Norenzayan, A., Smith, E. E., Kim, B. J., & Nisbett, R. E. (2002). Cultural preferences for formal versus intuitive reasoning. Cognitive Science, 26, 653-684.

varfun.Risen.1 *varfun.Risen.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#18_risen

Usage

```
varfun.Risen.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Variables

rise1.3=likelihood that the professor will call on you (unprepared condition); rise2.3=likelihood that the professor will call on you (prepared condition); for both, higher numbers=higher likelihood Variable = "ex.subjp" which asked if participants were recruited through a university subject pool. 1 = yes, 2 = no.

References

Risen, J. L., & Gilovich, T. (2008). Why people are reluctant to tempt fate. **Journal of Personality and Social Psychology**, 95, 293.

varfun.Risen.2 *varfun.Risen.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#18_risen

Usage

```
varfun.Risen.2(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

rise1.3=likelihood that the professor will call on you (unprepared condition); rise2.3=likelihood that the professor will call on you (prepared condition);

for both, higher numbers=higher likelihood All participants that answer the dependent measure will be included in analysis. The primary confirmatory test for comparing the original and replication effect size will be based on only the samples using undergraduate students.

References

Risen, J. L., & Gilovich, T. (2008). Why people are reluctant to tempt fate. Journal of Personality and Social Psychology, 95, 293.

---

varfun.Ross.1 *varfun.Ross.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#13_ross1

Usage

```
varfun.Ross.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

ross.s1.1 = percentage of peers; ross.s1.2 = you; values: 1=sign; 2=refuse

---

varfun.Ross.2 *varfun.Ross.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#14_ross2

Usage

```
varfun.Ross.2(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

ross.s2.1=percentage of peers; ross.s2.2=you; values: 1=Pay; 2=Appear in court

---

varfun.Rottenstreich.1 *varfun.Rottenstreich.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#5_rottenstreich

Usage

```
varfun.Rottenstreich.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

---

varfun.Savani.1 *varfun.Savani.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#19_savani

Usage

```
varfun.Savani.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

sava1.N=interpersonal actions;

sava2.N=personal actions;

'sava1.4', 'sava1.5', 'sava1.9', 'sava1.10', 'sava1.15', 'sava1.16' , 'sava1.21', 'sava1.22', 'sava1.27', 'sava1.28', 'sava1.33', 'sava1.34', 'sava1.38', 'sava1.39', 'sava1.43', 'sava1.44'

sava1.4=choice (buy a gift; 1=choice; 2=no choice);

sava1.5=importance (buy a gift);

sava1.9=choice (take a friend at the restaurant; 1=choice; 2=no choice);

sava1.10=importance (restaurant);

sava1.15=choice (trip; 1=choice, 2 and 3 = no choice);

sava1.16=importance (trip);

sava1.21=choice (dinner; 1=choice, 2 and 3 = no choice);

sava1.22=importance (dinner);

sava1.27=choice (errand; 1=choice, 2 and 3 = no choice);

sava1.28=importance (errand);

sava1.33=choice (help, 1=choice, 2 & 3 = no choice);

sava1.34=importance (help);

sava1.38=choice (advice, 1=choice, 2 & 3 = no choice);

sava1.39=importance (advice);

sava1.43=choice (friends, 1=choice, 2 & 3 = no choice);

sava1.44=importance (friends);

sava2.4=choice (buy for yourself; 1=choice; 2=no choice);

sava2.5=importance (buy for yourself);

sava2.9=choice (at the restaurant by yourself; 1=choice; 2=no choice);

sava2.10=importance (restaurant by yourself);

sava2.15=choice (trip alone; 1=choice, 2 and 3 = no choice);

sava2.16=importance (trip alone);

sava2.21=choice (out for dinner; 1=choice, 2 and 3 = no choice);

sava2.22=importance (out for dinner);

sava2.27=choice (errand for yourself; 1=choice, 2 and 3 = no choice);

sava2.28=importance (errand for yourself);

sava2.33=choice (ask for help, 1=choice, 2 & 3 = no choice);

sava2.34=importance (ask for help);

sava2.38=choice (take a course, 1=choice, 2 & 3 = no choice);

sava2.39=importance (take a course);

sava2.43=choice (friends, 1=choice, 2 & 3 = no choice);

sava2.44=importance (friends);

For all importance items: higher numbers=higher importance

we will only include university data collections in the primary confirmatory analysis to be compared with the original effect sizes.

Data for all participants will be included to examine variability across sample and setting. However, participants must respond to all choice and importance of choice questions to be included in the analysis.

`varfun.Schwarz.1` *varfun.Schwarz.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#24_schwarz

Usage

```
varfun.Schwarz.1(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Variables

schw1.1=life sat (first); schw1.2=partner satisfaction (second); schw2.1=partner satisfaction (first); schw2.2=life sat (second). for all, higher numbers=higher satisfaction

References

Schwarz, N., Strack, F., & Mai, H. P. (1991). Assimilation and contrast effects in part-whole question sequences: A conversational logic analysis. **Public Opinion Quarterly, 55**, 3-23.

`varfun.Schwarz.2` *varfun.Schwarz.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#24_schwarz

Usage

```
varfun.Schwarz.2(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Details

Analysis plan: We will compute the correlation between responses to the general and specific question in each item order condition, and then compare the correlations using the Fisher r-to-z transformation. Participants with valid responses to both items will be included in the analysis.

Value

Dataset ready for analysis

References

Schwarz, N., Strack, F., & Mai, H. P. (1991). Assimilation and contrast effects in part-whole question sequences: A conversational logic analysis. **Public Opinion Quarterly, 55**, 3-23.

---

varfun.Shafir.1 *varfun.Shafir.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#25_shafir

CHANGED The analysis in the orignal article was likely as follows:

- Count number of Parent B choices in both conditions - Sum the proportions - Divide by 2 and test against proportion = .5

Usage

```
varfun.Shafir.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

shaf1.1=choice (award condition; Parent A=1, Parent B=2); shaf2.1=choice (deny condition; Parent A=1, Parent B=2)

References

Shafir, E. (1993). Choosing versus rejecting: Why some options are both better and worse than others. Memory & Cognition, 21, 546-556.

varfun.Tversky.1 *varfun.Tversky.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#16_tversky

Usage

```
varfun.Tversky.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

tver1.1=choice ($250 wall hanging condition, yes=1, no=2); tver2.1=choice ($30 wall hanging cond, yes=1, no=2).

Materials and procedure. Participants will receive one of two scenarios from the original with dollar amounts approximately adjusted for inflation and the consumer items being replaced with a ceramic vase and a wall hanging.

tver1.1 Imagine that you are about to purchase a ceramic vase for $30, and a wall hanging for $250. The salesman informs you that the wall hanging you wish to buy is on sale for $240 at the other branch of the store, located 20 minutes drive away. Would you make the trip to the other store? m Yes, I would go to the other branch. (1) m No, I would not go to the other branch. (2)

tver2.1 Imagine that you are about to purchase a ceramic vase for $250, and a wall hanging for $30. The salesman informs you that the wall hanging you wish to buy is on sale for $20 at the other branch of the store, located 20 minutes drive away. Would you make the trip to the other store? m Yes, I would go to the other branch. (1) m No, I would not go to the other branch. (2)

References

Tversky, A., Kahneman, D. (1981). The framing of decisions and the psychology of choice. Science, 211, 453-458.

varfun.vanLange.1 *van.Lange.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#vanlange

https://manylabsopenscience.github.io/ML2_PoPS_proposal#10_van_lange

Usage

```
varfun.vanLange.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis A list object containing fields **SVO**, **Siblings** and **N**

Variables

van.p.1.2_1 TO van.p.1.2_6 are the items of SVO measure.

murphy et al. (2011) scoring: SVO degress=arctan [(mean Alloc other - 50)/(mean Allocation self - 50)].

See SVO codes (this doc) for the list of paired amounts van.p2.1_1_TEXT= # of older siblings; van.p2.1_2_TEXT= # of younger siblings.

varfun.Zaval.1 *varfun.Zaval.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#26_zaval

Usage

```
varfun.Zaval.1(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

zav1.1 TO zav1.13 provide COLD primes. zav2.1 TO zav2.13 provide HEAT primes. zav.dv.2=belief; zav.dv.3=concern; higher numbers=higher belief/concern.

References

Zaval, L., Keenan, E. A., Johnson, E. J., & Weber, E. U. (2014). How warm days increase belief in global warming. **Nature Climate Change**, 4, 143-147.

`varfun.Zhong.1` *varfun.Zhong.1*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#23_zhong

Usage

```
varfun.Zhong.1(vars)
```

Arguments

`vars`	A list object generated by `get.sourceData` containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

zhon1.1= unethical condition; zhon2.1=ethical condition;

zhon.dv.1_1 TO zhon.dv.1_10=desirability of products (both conditions); higher numbers=higher desirability;

Products list:

Clean: Dove shower soap (zhon.dv.1_2), Crest toothpaste (zhon.dv.1_3), Windex glass cleaner (zhon.dv.1_7), Lysol countertop disinfectant (zhon.dv.1_8), Tide laundry detergent (zhon.dv.1_10)

Not-clean: Post-it notes (zhon.dv.1_1), Nantucket Nectars juice (zhon.dv.1_4), Energizer batteries (zhon.dv.1_5), Sony cd cases (zhon.dv.1_6), Snickers candy bar (zhon.dv.1_9),

References

Zhong, C. B., & Liljenquist, K. (2006). Washing away your sins: Threatened morality and physical cleansing. Science, 313, 1451???1452.

varfun.Zhong.2 *varfun.Zhong.2*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#23_zhong

Usage

```
varfun.Zhong.2(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

zhon1.1= unethical condition; zhon2.1=ethical condition;

zhon.dv.1_1 TO zhon.dv.1_10=desirability of products (both conditions); higher numbers=higher desirability;

Products list:

Clean: Dove shower soap (zhon.dv.1_2), Crest toothpaste (zhon.dv.1_3), Windex glass cleaner (zhon.dv.1_7), Lysol countertop disinfectant (zhon.dv.1_8), Tide laundry detergent (zhon.dv.1_10)

Not-clean: Post-it notes (zhon.dv.1_1), Nantucket Nectars juice (zhon.dv.1_4), Energizer batteries (zhon.dv.1_5), Sony cd cases (zhon.dv.1_6), Snickers candy bar (zhon.dv.1_9),

References

Zhong, C. B., & Liljenquist, K. (2006). Washing away your sins: Threatened morality and physical cleansing. Science, 313, 1451–1452.

varfun.Zhong.3 *varfun.Zhong.3*

Description

https://manylabsopenscience.github.io/ML2_PoPS_proposal#23_zhong

Usage

```
varfun.Zhong.3(vars)
```

Arguments

vars	A list object generated by get.sourceData containing cleaned data and variable labels.

Value

Dataset ready for analysis

Variables

zhon1.1= unethical condition; zhon2.1=ethical condition;

zhon.dv.1_1 TO zhon.dv.1_10=desirability of products (both conditions); higher numbers=higher desirability;

Products list:

Clean: Dove shower soap (zhon.dv.1_2), Crest toothpaste (zhon.dv.1_3), Windex glass cleaner (zhon.dv.1_7), Lysol countertop disinfectant (zhon.dv.1_8), Tide laundry detergent (zhon.dv.1_10)

Not-clean: Post-it notes (zhon.dv.1_1), Nantucket Nectars juice (zhon.dv.1_4), Energizer batteries (zhon.dv.1_5), Sony cd cases (zhon.dv.1_6), Snickers candy bar (zhon.dv.1_9),

References

Zhong, C. B., & Liljenquist, K. (2006). Washing away your sins: Threatened morality and physical cleansing. Science, 313, 1451???1452.

---

vioQtile *vioQtile*

Description

vioQtile

Usage

```
vioQtile(gg = NULL, qtiles = NULL, probs = seq(0, 1, 0.25),
  labels = paste(probs[-1] * 100), withData = FALSE)
```

Arguments

gg	A ggplot.
qtiles	Quantiles.
probs	Probabilities.
labels	Labels.
withData	Return Data.

Details

This is adapted from: http://stackoverflow.com/questions/22278951/combining-violin-plot-with-box-plo

Changed: Deal with 'empty' quantile groups Deal with original data More input, more output

`%0!0%` *Rose tinted infix*

Description

When your functions wear these rose tinted glasses, the world will appear to be a nicer, fluffier place.

Usage

```
x %0!0% y
```

Arguments

`x`	If `x` is any of `Inf,-Inf,NA,NaN,NULL,length(x)==0`, it will return `y`; otherwise it returns `x`.
`y`	The value to return in case of catastrophy `>!<`

Author(s)

Fred Hasselman

See Also

purrr::

Examples

```
Inf %0!0% NA

numeric(0) %0!0% ''

NA %0!0% 0

NaN %0!0% NA

NULL %0!0% NA
```

Index