Package ‘tidycomm’

October 14, 2022

Title Data Modification and Analysis for Communication Research

Version 0.2.1

Description Provides convenience functions for common data
modification and analysis tasks in communication research. This
includes functions for univariate and bivariate data analysis, index
generation and reliability computation, and intercoder reliability
tests. All functions follow the style and syntax of the tidyverse, and
are construed to perform their computations on multiple variables at
once. Functions for univariate and bivariate data analysis comprise
summary statistics for continuous and categorical variables, as well
as several tests of bivariate association including effect sizes.
Functions for data modification comprise index generation and
automated reliability analysis of index variables. Functions for
intercoder reliability comprise tests of several intercoder
reliability estimates, including simple and mean pairwise percent
agreement, Krippendorff's Alpha (Krippendorff 2004, ISBN:
9780761915454), and various Kappa coefficients (Brennan & Prediger

License GPL-3

URL https://joon-e.github.io/tidycomm/

BugReports https://github.com/joon-e/tidycomm/issues

Depends R (>= 2.10)

Imports broom, dplyr, forcats, glue, magrittr, MBESS, purrr, rlang,
stringr, tibble, tidy

Suggests covr, knitr, rmarkdown, testthat (>= 2.1.0), tidyselect

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no
add_index

Description

Add a rowwise mean or sum index of specific variables to the dataset.

Usage

add_index(data, name, ..., type = "mean", na.rm = TRUE, cast.numeric = FALSE)

Arguments

data         a tibble
name         Name of the index column to compute.
...           Variables used for the index.
type         Type of index to compute. Either "mean" (default) or "sum".
na.rm         a logical value indicating whether NA values should be stripped before the computation proceeds. Defaults to TRUE.
cast.numeric  a logical value indicating whether all variables selected for index computation should be converted to numeric. Useful if computing indices from factor variables. Defaults to FALSE.
correlate

Value

a tibble

See Also

generate_reliability() to compute reliability estimates of added index variables.

Examples

WoJ %>% add_index(ethical_flexibility, ethics_1, ethics_2, ethics_3, ethics_4)
WoJ %>% add_index(ethical_flexibility, ethics_1, ethics_2, ethics_3, ethics_4, type = "sum")

correlate(data, ..., method = "pearson")

data

a tibble

... Variables to compute correlations for (column names). Leave empty to compute for all numeric variables in data.

method

a character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman"

Value

a tibble

Examples

WoJ %>% correlate(ethics_1, ethics_2, ethics_3)
WoJ %>% correlate()
crosstab  

**Crosstab variables**

**Description**
Computes contingency table for one independent (column) variable and one or more dependent (row) variables.

**Usage**
```r
crosstab(
  data,
  col_var,
  ..., 
  add_total = FALSE,
  percentages = FALSE,
  chi_square = FALSE
)
```

**Arguments**
- `data` a tibble
- `col_var` Independent (column) variable.
- `...` Dependent (row) variables.
- `add_total` Logical indicating whether a 'Total' column should be computed. Defaults to FALSE.
- `percentages` Logical indicating whether to output column-wise percentages instead of absolute values. Defaults to FALSE.
- `chi_square` Logical indicating whether a Chi-square test should be computed. Test results will be reported via `message()`. Defaults to FALSE.

**Value**
a tibble

**See Also**
Other categorical: `tab_frequencies()`

**Examples**
```r
WoJ %>% crosstab(reach, employment)
WoJ %>% crosstab(reach, employment, add_total = TRUE, percentages = TRUE, chi_square = TRUE)
```
**describe**  
*Describe numeric variables*

**Description**
Describe numeric variables by several measures of central tendency and variability. If no variables are specified, all numeric (integer or double) variables are described.

**Usage**
describe(data, ..., na.rm = TRUE)

**Arguments**
- data: a tibble
- ... Variables to describe (column names). Leave empty to describe all numeric variables in data.
- na.rm: a logical value indicating whether NA values should be stripped before the computation proceeds. Defaults to TRUE.

**Value**
a tibble

**See Also**
Other descriptives: describe_cat()

**Examples**

```r
iris %>% describe()
mtcars %>% describe(mpg, am, cyl)
```

---

**describe_cat**  
*Describe categorical variables*

**Description**
Describe categorical variables by N, number of unique values, and mode. Note that in case of multiple modes, the first mode by order of values is chosen.

**Usage**
describe_cat(data, ...)

---
Arguments

- `data`  
a tibble

... Variables to describe (column names). Leave empty to describe all categorical variables in data.

Details

If no variables are specified, all categorical (character or factor) variables are described.

Value

a tibble

See Also

Other descriptives: `describe()`

Examples

```r
iris %>% describe_cat()
```

fbposts  
Facebook posts reliability test

Description

45 political Facebook posts coded by 6 coders for an intercoder reliability test, focused on populist messages.

Usage

`fbposts`

Format

A data frame with 270 rows and 7 variables

- `post_id`  
Numeric id of the coded Facebook post

- `coder_id`  
Numeric id of the coder

- `type`  
Type of Facebook post, one of "link", "photo", "status", or "video"

- `n_pictures`  
Amount of pictures attached to the post, ranges from 0 to 6

- `pop_elite`  
Populism indicator: Does the Facebook post attack elites?, 0 = "no attacks on elites", 1 = "attacks political actors", 2 = "attacks public administration actors", 3 = "attacks economical actors", 4 = "attacks media actors/journalists", 9 = "attacks other elites"
get_reliability

pop_people  Populism indicator: Does the Facebook refer to 'the people'? 0 = "does not refer to 'the people'" 1 = "refers to 'the people"

pop_othering  Populism indicator: Does the Facebook attack 'others'? 0 = "no attacks on 'others'" 1 = "attacks other cultures" 2 = "attacks other political stances" 3 = "attacks other 'others'"

get_reliability  Get reliability estimates of index variables

Description
Get reliability estimates of index variables created with add_index.

Usage
get_reliability(
  data,
  ..., 
  type = "alpha",
  interval.type = NULL,
  bootstrap.samples = NULL,
  conf.level = NULL,
  progress = FALSE
)

Arguments
- data  a tibble
- ...  Index variables created with add_index. Leave empty to get reliability estimates for all index variables.
- type  Type of reliability estimate. See ci.reliability
- interval.type  Type of reliability estimate confidence interval. See ci.reliability
- bootstrap.samples  Number of bootstrap samples for CI calculation. See ci.reliability
- conf.level  Confidence level for estimate CI. See ci.reliability
- progress  Show progress for reliability estimate computation. Useful if using computationally intense computations (e.g., many bootstrapping samples) and many index variables.

Value
a tibble

See Also
add_index() to create index variables
Examples

WoJ %>%
  add_index(ethical_flexibility, ethics_1, ethics_2, ethics_3, ethics_4) %>%
  get_reliability()

---

**tab_frequencies**

**Tabulate frequencies**

Description

Tabulates frequencies for one or more categorical variable, including relative, and cumulative frequencies.

Usage

`tab_frequencies(data, ...)`

Arguments

- `data` a tibble
- `...` Variables to tabulate

Value

a tibble

See Also

Other categorical: `crosstab()`

Examples

WoJ %>% tab_frequencies(employment)
WoJ %>% tab_frequencies(employment, country)
**Description**

Performs an intercoder reliability test by computing various intercoder reliability estimates for the included variables.

**Usage**

```r
test_icr(
  data,  # a tibble
  unit_var,  # Variable with unit identifiers
  coder_var,  # Variable with coder identifiers
  ...,  # Variables to compute intercoder reliability estimates for. Leave empty to compute for all variables (excluding `unit_var` and `coder_var`) in data.
  levels = NULL,  # Optional named vector with levels of test variables
  na.omit = FALSE,  # Logical indicating whether NA values should be stripped before computation. Defaults to FALSE.
  agreement = TRUE,  # Logical indicating whether simple percent agreement should be computed. Defaults to TRUE.
  holsti = TRUE,  # Logical indicating whether Holsti’s reliability estimate (mean pairwise agreement) should be computed. Defaults to TRUE.
  kripp_alpha = TRUE,  # Logical indicating whether Krippendorff’s Alpha should be computed. Defaults to TRUE.
  cohens_kappa = FALSE,  # Logical indicating whether Cohen’s Kappa should be computed. Defaults to FALSE.
  fleiss_kappa = FALSE,  # Logical indicating whether Fleiss’s Kappa should be computed. Defaults to FALSE.
  brennan_prediger = FALSE,  # Logical indicating whether Brennan and Prediger’s Kappa should be computed. Defaults to FALSE.
  lotus = FALSE,  # Logical indicating whether Lotus’s Kappa should be computed. Defaults to FALSE.
  s_lotus = FALSE  # Logical indicating whether S-Lotus’s Kappa should be computed. Defaults to FALSE.
)
```

**Arguments**

- `data`: a tibble
- `unit_var`: Variable with unit identifiers
- `coder_var`: Variable with coder identifiers
- `...`: Variables to compute intercoder reliability estimates for. Leave empty to compute for all variables (excluding `unit_var` and `coder_var`) in data.
- `levels`: Optional named vector with levels of test variables
- `na.omit`: Logical indicating whether NA values should be stripped before computation. Defaults to FALSE.
- `agreement`: Logical indicating whether simple percent agreement should be computed. Defaults to TRUE.
- `holsti`: Logical indicating whether Holsti’s reliability estimate (mean pairwise agreement) should be computed. Defaults to TRUE.
- `kripp_alpha`: Logical indicating whether Krippendorff’s Alpha should be computed. Defaults to TRUE.
- `cohens_kappa`: Logical indicating whether Cohen’s Kappa should be computed. Defaults to FALSE.
fleiss_kappa Logical indicating whether Fleiss’ Kappa should be computed. Defaults to FALSE.
brennan_prediger Logical indicating whether Brennan & Prediger’s Kappa should be computed (extension to 3+ coders as proposed by von Eye (2006)). Defaults to FALSE.
lotus Logical indicating whether Fretwurst’s Lotus should be computed. Defaults to FALSE
s_lotus Logical indicating whether Fretwurst’s standardized Lotus (S-Lotus) should be computed. Defaults to FALSE.

Value
a tibble

References

Examples
```
fbposts %>% test_icr(post_id, coder_id, pop_elite, pop_othering)
fbposts %>% test_icr(post_id, coder_id, levels = c(n_pictures = "ordinal"), fleiss_kappa = TRUE)
```

Description
Turns the tibble exported from correlate into a correlation matrix.

Usage
to_correlation_matrix(data)
**t_test**

**Arguments**

- `data` a tibble returned from `correlate`

**Value**

a tibble

**Examples**

WoJ %>% correlate() %>% to_correlation_matrix()

---

**t_test**

*Compute t-tests*

**Description**

Computes t-tests for one group variable and specified test variables. If no variables are specified, all numeric (integer or double) variables are used.

**Usage**

`t_test(
  data,
  group_var,
  ..., 
  var.equal = TRUE,
  paired = FALSE,
  pooled_sd = TRUE,
  levels = NULL,
  case_var = NULL
)`

**Arguments**

- `data` a tibble
- `group_var` group variable (column name)
- `...` test variables (column names). Leave empty to compute t-tests for all numeric variables in data.
- `var.equal` a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used. Defaults to TRUE.
- `paired` a logical indicating whether you want a paired t-test. Defaults to FALSE.
- `pooled_sd` a logical indicating whether to use the pooled standard deviation in the calculation of Cohen’s d. Defaults to TRUE.
levels optional: a vector of length two specifying the two levels of the group variable.

Case_var optional: case-identifying variable (column name). If you set paired = TRUE, specifying a case variable will ensure that data are properly sorted for a dependent t-test.

Value

a tibble

Examples

WoJ %>% t_test(temp_contract, autonomy_selection, autonomy_emphasis)
WoJ %>% t_test(temp_contract)
WoJ %>% t_test(employment, autonomy_selection, autonomy_emphasis,
levels = c("Full-time", "Freelancer"))

unianova

Compute one-way ANOVAs

Description

Computes one-way ANOVAs for one group variable and specified test variables. If no variables are specified, all numeric (integer or double) variables are used.

Usage

unianova(data, group_var, ..., descriptives = FALSE, post_hoc = FALSE)

Arguments

data a tibble

group_var group variable (column name)

... test variables (column names). Leave empty to compute ANOVAs for all numeric variables in data.

Descriptives a logical indicating whether descriptive statistics (mean & standard deviation) for all group levels should be added to the returned tibble. Defaults to FALSE.

Post_hoc a logical indicating whether post-hoc tests (Tukey's HSD) should be computed. Results of the post-hoc test will be added in a list column of result tibbles.

Value

a tibble
Examples

WoJ %>% unianova(employment, autonomy_selection, autonomy_emphasis)
WoJ %>% unianova(employment)
WoJ %>% unianova(employment, descriptives = TRUE, post_hoc = TRUE)

Description

A subset of data from the Worlds of Journalism 2012-16 study containing survey data of 1,200 journalists from five European countries.

Usage

WoJ

Format

A data frame with 1200 rows and 15 variables:

country  Country of residence
reach  Reach of medium
employment  Current employment situation
temp_contract  Type of contract (if current employment situation is either full-time or part-time
autonomy_selection  Autonomy in news story selection, scale from 1 (no freedom at all) to 5 (complete freedom)
autonomy_emphasis  Autonomy in news story emphasis, scale from 1 (no freedom at all) to 5 (complete freedom)
ethics_1  Agreement with statement "Journalists should always adhere to codes of professional ethics, regardless of situation and context", scale from 1 (strongly disagree) to 5 (strongly agree) (reverse-coded!)
ethics_2  Agreement with statement "What is ethical in journalism depends on the specific situation.", scale from 1 (strongly disagree) to 5 (strongly agree)
ethics_3  Agreement with statement "What is ethical in journalism is a matter of personal judgment.", scale from 1 (strongly disagree) to 5 (strongly agree)
ethics_4  Agreement with statement "It is acceptable to set aside moral standards if extraordinary circumstances require it.", scale from 1 (strongly disagree) to 5 (strongly agree)
work_experience  Work experience as a journalist in years
trust_parliament  Trust placed in parliament, scale from 1 (no trust at all) to 5 (complete trust)
trust_government  Trust placed in government, scale from 1 (no trust at all) to 5 (complete trust)
trust_parties  Trust placed in parties, scale from 1 (no trust at all) to 5 (complete trust)
trust_politicians  Trust placed in politicians in general, scale from 1 (no trust at all) to 5 (complete trust)
Source

Index

* ANOVA
  unianova, 12
* categorical
crosstab, 4
tab_frequencies, 8
* correlations
correlate, 3
* correlation
to_correlation_matrix, 10
* datasets
  fbposts, 6
  WoJ, 13
* descriptives
describe, 5
describe_cat, 5
* intercoder reliability
test_icr, 9
* reliability
get_reliability, 7
* t-test
t_test, 11
  add_index, 2, 7
  add_index(), 7
  ci.reliability, 7
correlate, 3, 10, 11
crosstab, 4, 8
describe, 5, 6
describe_cat, 5, 5
fbposts, 6
get_reliability, 7
get_reliability(), 3
t_test, 11
tab_frequencies, 4, 8
test_icr, 9
tibble, 2–12