Package 'scaledescr'

March 13, 2026

Type Package

Title Descriptive, Reliability, and Inferential Tables for Psychometric Scales and Demographic Data

Version 0.2.2

Description
Provides functions to format and summarise already computed outputs from commonly used statistical and psychometric functions into compact, single-row tables and simple graphs, with utilities to export results to CSV, Word, and Excel formats. The package does not implement new statistical methods or estimation procedures; instead, it organises and presents results obtained from existing functions such as psych::describe(), psych::alpha(), stats::t.test(), and gtsummary::tbl_summary() to streamline reporting workflows in clinical and psychological research.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports dplyr, gtsummary, purrr, rlang, stats, officer, openxlsx, utils, psych, stringr, flextable

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Darshankumar Dharaiya [aut, cre] (ORCID: <https://orcid.org/0009-0009-9351-8602>)

Maintainer Darshankumar Dharaiya <dharaiya.darshan@gmail.com>

Depends R (>= 4.1.0)

Repository CRAN

Date/Publication 2026-03-13 17:00:02 UTC

Contents

`make_alpha_table` *Wrap a pre computed psych::alpha object into a single row table*

Description

Wrap a pre computed psych::alpha object into a single row table

Usage

```
make_alpha_table(alpha_res, scale_name = "Scale")
```

Arguments

`alpha_res`	A psych::alpha object (already computed)
`scale_name`	Name of the scale (default: "Scale")

Value

A data frame with columns: Scale, 95

Examples

```
# Create a minimal "psych::alpha" like object manually
alpha_obj <- list(
  total = list(
    raw_alpha = 0.85,
    lower = 0.78,
    upper = 0.92
  )
)

# Generate the formatted alpha table
make_alpha_table(alpha_obj, scale_name = "PHQ-9")
```

`make_chisq_test_table` *Create a one-row summary table for a chi-square test*

Description

This function formats the result of a pre-computed `chisq.test()` into a single-row, report-ready data frame. It supports goodness-of-fit, independence, and homogeneity chi-square tests and includes an appropriate effect size with a qualitative interpretation.

Usage

```
make_chisq_test_table(
  chisq_object,
  test_type = c("goodness-of-fit", "independence", "homogeneity")
)
```

Arguments

`chisq_object`	An object of class `"htest"` produced by `stats::chisq.test()`.
`test_type`	Character string specifying the type of chi-square test. One of `"goodness-of-fit"`, `"independence"`, or `"homogeneity"`.

Details

The function does not perform the chi-square test itself and does not introduce new statistical methods. All test statistics are extracted directly from the supplied `chisq.test()` object.

For goodness-of-fit tests, Cohen's w is reported. For tests of independence and homogeneity, Cramér's V is reported. Effect size interpretations follow conventional benchmarks (0.10 = small, 0.30 = medium, 0.50 = large).

Value

A single-row data frame with the following columns:

- `test`: Type of chi-square test
- `chi_square`: Chi-square statistic
- `df`: Degrees of freedom
- `p_value`: p-value
- `N`: Total sample size
- `effect_size`: Effect size (Cohen's w or Cramér's V)
- `effect_type`: Type of effect size reported
- `effect_interpretation`: Qualitative interpretation of effect size

Examples

```
# Goodness-of-fit example
observed <- c(40, 30, 50)
chisq_gof <- chisq.test(observed)

make_chisq_test_table(
  chisq_object = chisq_gof,
  test_type = "goodness-of-fit"
)

# Independence test example
tbl <- matrix(c(20, 30, 10, 40), nrow = 2)
chisq_ind <- chisq.test(tbl)

make_chisq_test_table(
  chisq_object = chisq_ind,
  test_type = "independence"
)
```

`make_dataframe_to_output`

Export a data frame to CSV, Word (.docx), or Excel (.xlsx) format.

Description

If no output path is specified, the file is written to a temporary directory using ‘tempdir()‘. For reproducible workflows, users are encouraged to explicitly specify an output location, either the current working directory or a full file path.

Usage

```
make_dataframe_to_output(data, filename = NULL, format = "csv", path = NULL)
```

Arguments

`data`	A data frame to export.
`filename`	Optional base file name (without extension). Defaults to object name. (without path and without extension). If not provided, the name of the input object is used.
`format`	Character string specifying the output format. One of ‘"csv"‘, ‘"word"‘, or ‘"excel"‘. Default is ‘"csv"‘.
`path`	Optional character string specifying the directory where the file should be saved. If ‘NULL‘ (default), the file is written to a temporary directory. Use ‘path = getwd()‘ to save the file in the current working directory, or provide a full directory path.

Details

- CSV files are written using ‘utils::write.csv()‘ - Word files (‘docx‘) are created using the ‘officer‘ package - Excel (‘xlsx‘) files are created using the ‘openxlsx‘ package

Value

Invisibly returns the full file path to the generated output file.

Examples

```
## Not run:

# Example dataset available in base R
data_df <- head(mtcars)

#-------------------------------------------------
# 1. Simplest use: export to CSV in temp directory
#-------------------------------------------------
make_dataframe_to_output(data_df)

#-------------------------------------------------
# 2. Specify filename
#-------------------------------------------------
make_dataframe_to_output(
  data = data_df,
  filename = "mtcars_sample"
)

#-------------------------------------------------
# 3. Export as Word document
#-------------------------------------------------
make_dataframe_to_output(
  data = data_df,
  filename = "mtcars_word_table",
  format = "word"
)

#-------------------------------------------------
# 4. Export as Excel file
#-------------------------------------------------
make_dataframe_to_output(
  data = data_df,
  filename = "mtcars_excel_table",
  format = "excel"
)

#-------------------------------------------------
# 5. Save to current working directory
#-------------------------------------------------
make_dataframe_to_output(
  data = data_df,
  filename = "mtcars_current_folder",
```

```
  format = "csv",
  path = "getwd()"
)

#------------------------------------------------
# 6. Save Excel file to current working directory
#------------------------------------------------
make_dataframe_to_output(
  data = data_df,
  filename = "mtcars_excel_current",
  format = "excel",
  path = "getwd()"
)

#------------------------------------------------
# 7. Export another base dataset (iris)
#------------------------------------------------
make_dataframe_to_output(
  data = head(iris),
  filename = "iris_sample",
  format = "word"
)

#------------------------------------------------
# 8. Using a custom folder path
#------------------------------------------------
make_dataframe_to_output(
  data = head(airquality),
  filename = "airquality_data",
  format = "excel",
  path = "D:/output_folder"
)


## End(Not run)
```

`make_demographic_table` *Create a demographics summary table*

Description

Create a demographics summary table

Usage

```
make_demographic_table(data, vars, continuous_vars = NULL)
```

Arguments

data	A data frame
vars	demographic variables to include in the table
continuous_vars	Optional subset of vars to be treated as continuous

Value

A gtsummary table

Examples

```
df <- data.frame(
  age = c("25", "30 years", "35", " 40 ", "22.5", "28+", NA, ""),
  sex = c("M", "F", "m", "f", " M ", "F", "m", NA),
  education = c("HS", "BA", "MA", "ma", "Hs", "Ma", "Ba Bed", "Msc bed ")
)

# Generate a demographic summary table (assign to object to avoid printing)
demo_table <- make_demographic_table(df, vars = c("age", "sex", "education"))
demo_table # optionally inspect the table
```

---

make_demographic_table_to_output

Create and Export Demographic Summary Table

Description

Generates a demographic summary table using ‘make_demographic_table()‘ and exports the resulting table to Word, Excel, or CSV format.

This is a convenience wrapper that combines analysis and output in a single step for users who want immediate file export.

Usage

```
make_demographic_table_to_output(
  data,
  vars,
  continuous_vars = NULL,
  file_name = "demographic_table",
  format = c("word", "excel", "csv"),
  path = NULL
)
```

Arguments

`data`	A data frame containing the dataset.
`vars`	Character vector of variable names to include in the demographic summary.
`continuous_vars`	Optional character vector specifying which variables should be treated as continuous. If ‘NULL‘, numeric variables are automatically detected.
`file_name`	Name of the output file WITHOUT extension (e.g., "demographics").
`format`	Output format. Must be one of: • ‘"word"‘ - exports to a Word document (.docx) • ‘"excel"‘ - exports to an Excel file (.xlsx) • ‘"csv"‘ - exports to a CSV file (.csv)
`path`	Optional character string specifying the directory where the file should be saved. If ‘NULL‘ (default), the file is written to a temporary directory. Provide a full directory path or use getwd() during the function call.

Details

The demographic table includes:

- Categorical variables reported as n (
- Continuous variables reported as Mean (SD)

Word export uses ‘flextable‘ and ‘officer‘ for formatting. Excel export uses ‘openxlsx‘. CSV export uses base R ‘write.csv()‘.

Value

Invisibly returns the full file path to the generated output file.

`make_independent_t_test_table` *Create a one-row summary table for an independent-samples t-test*

Description

This function performs an independent-samples t-test (Welch’s t-test by default) between two groups defined by a binary grouping variable and returns a single-row, report-ready data frame. The output includes group names, sample sizes, mean difference, test statistics, p-value, and effect size (Cohen’s d) with a qualitative interpretation.

Usage

```
make_independent_t_test_table(data, outcome, group)
```

Arguments

data	A data frame containing the outcome and grouping variables.
outcome	Character string specifying the numeric outcome variable.
group	Character string specifying the grouping variable. Must have exactly two levels.

Details

The function is intended for streamlined reporting and does not introduce new statistical methods. All computations rely on `stats::t.test()`.

Welch's t-test is used by default, which does not assume equal variances. Cohen's d is computed using the pooled standard deviation for comparability with conventional benchmarks. Group ordering follows the factor level order of the grouping variable.

Value

A single-row data frame with the following columns:

- `test`: Name of the statistical test
- `group1`, `group2`: Group labels
- `mean_diff`: Mean difference between groups (group1 - group2)
- `t_value`: t statistic
- `df`: Degrees of freedom
- `p_value`: p-value
- `n_group1`, `n_group2`: Sample sizes per group
- `cohens_d`: Cohen's d effect size
- `interpretation`: Qualitative interpretation of effect size

Examples

```
set.seed(123)

data_t <- data.frame(
  group = rep(c("CBT", "Psychodynamic"), each = 30),
  score = c(
    rnorm(30, mean = 18, sd = 4),
    rnorm(30, mean = 21, sd = 4)
  )
)

make_independent_t_test_table(
  data = data_t,
  outcome = "score",
  group = "group"
)
```

`make_paired_t_test_table` *Create a one-row summary table of a paired t-test*

Description

This function performs a paired t-test between two numeric variables in a data frame and returns a one-row summary table including means, mean difference, t-value, degrees of freedom, p-value, and confidence interval.

Usage

```
make_paired_t_test_table(
  data,
  var1,
  var2,
  var_name = NULL,
  alternative = "two.sided",
  conf.level = 0.95
)
```

Arguments

`data`	A data frame containing the two numeric variables.
`var1`	Character string. Name of the first variable (observation 1) in ‘data‘.
`var2`	Character string. Name of the second variable (observation 2) in ‘data‘.
`var_name`	Optional character string. Custom name for the variable to display in the table. Default is ‘var1 vs var2‘.
`alternative`	Character string specifying the alternative hypothesis. One of ‘"two.sided"‘, ‘"less"‘, or ‘"greater"‘. Default is ‘"two.sided"‘.
`conf.level`	Confidence level for the interval. Default is 0.95.

Value

A one-row data frame with columns:

- ‘Variable‘ - variable name
- ‘Mean_obs1‘ - mean of observation 1
- ‘Mean_obs2‘ - mean of observation 2
- ‘Mean_diff‘ - mean difference (obs1 - obs2)
- ‘t_value‘ - t statistic
- ‘df‘ - degrees of freedom
- ‘p_value‘ - p-value
- ‘CI_lower‘ - lower bound of confidence interval
- ‘CI_upper‘ - upper bound of confidence interval

Examples

```
# example data
df <- data.frame(
  before = c(10, 12, 14, 15, 11),
  after  = c(11, 13, 13, 16, 12)
)

# Run the paired t-test summary
make_paired_t_test_table(df, var1 = "before", var2 = "after")
```

`make_scale_description_table`

Create a Descriptive Statistics Table Row

Description

Computes and formats descriptive statistics for a scale total score into a single-row data frame suitable for reporting.

Usage

```
make_scale_description_table(x, scale_name, type = NULL)
```

Arguments

`x`	A numeric vector representing total scores of a scale.
`scale_name`	A single character string specifying the name of the scale.
`type`	Optional character string. If NULL (default), descriptive statistics are computed using `psych::describe()`. If set to `"summary"`, statistics are computed using `base::summary()`.

Details

This function is intended for reporting descriptive statistics of total scale scores, for which descriptive statistics are computed internally using `psych::describe()` or `base::summary()`.

Value

A single-row data frame with formatted descriptive statistics.

Examples

```
{
  phq9_data <- as.data.frame(matrix(sample(0:3, 10 * 9, replace = TRUE), 10, 9))
  colnames(phq9_data) <- paste0("Q", 1:9)
  phq9_data$total <- rowSums(phq9_data)

  make_scale_description_table(phq9_data$total, scale_name = "PHQ-9")
```

```
}
```

`scaledescr` *scaledescr*

Description

Provides helper functions to format and summarise already computed outputs from commonly used statistical and psychometric functions into compact, single-row tables and simple graphs. Functions such as make_scale_description_table(), make_demographic_table(), make_alpha_table(), make_paired_t_test_table(), and make_dataframe_to_output() organise results obtained from existing functions including psych::describe(), psych::alpha(), stats::t.test(), and gtsummary::tbl_summary() for streamlined reporting and export to CSV, Word, and Excel formats. The package does not implement new statistical methods or perform additional estimation.

Index