Package 'testflow'

November 3, 2026

Title A Workflow for Statistical Testing and Interpretation

Version 1.0.5

Author Imad El Badisy [aut, cre]

Maintainer Imad El Badisy <elbadisyimad@gmail.com>

Description Provides a unified workflow for choosing, running, and interpreting common statistical tests, from group comparisons and analysis of variance to regression, survival analysis, and diagnostic and agreement statistics. The package combines assumption checks, test selection, effect sizes, formatted results, plain-language interpretation, and a sample-size planning module covering continuous, binary, survival, ordinal, bioequivalence, and precision-based designs. Implemented methods follow standard references including Casella and Berger (2002, ISBN:9780534243128), Hollander et al. (2013, ISBN:9781118553299), Agresti (2013, ISBN:9780470463635), Cohen (1988, ISBN:9780805802832), Hosmer, Lemeshow and Sturdivant (2013, ISBN:9780470582473), and Julious (2010, ISBN:9781584887393).

URL https://CRAN.R-project.org/package=testflow

BugReports https://github.com/ielbadisy/testflow/issues

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

RoxygenNote 7.3.3

Imports stats, cli, rlang, basetable (>= 1.4.1), tidyselect, tibble, broom, ggplot2, survival, car

Suggests testthat (>= 3.0.0), knitr, rmarkdown, effectsize, patchwork, PowerTOST, irr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2026-11-03 00:10:02 UTC

Contents

as_tibble . . . 3
make_cardio_data . . . 3
plot.testflow . . . 4
print.summary.sample_size . . . 4
print.testflow . . . 5
print.testflow_sumtab . . . 5
report . . . 6
report_test . . . 6
sample_size . . . 7
sample_size_bioequivalence . . . 10
sample_size_cluster_adjust . . . 12
sample_size_precision . . . 13
summary.testflow . . . 17
sumtab . . . 17
test_agreement . . . 19
test_categorical . . . 20
test_correlation . . . 21
test_correlation_matrix . . . 22
test_cox . . . 23
test_diagnostic . . . 24
test_factorial . . . 25
test_groups . . . 26
test_icc . . . 27
test_linear_regression . . . 28
test_logistic_regression . . . 29
test_multinomial . . . 30
test_one_sample . . . 31
test_outliers . . . 32
test_paired . . . 33
test_paired_categorical . . . 34
test_proportion . . . 34
test_repeated . . . 35
test_repeated_categorical . . . 36
test_repeated_long . . . 37
test_roc . . . 38
test_survival . . . 40
test_two_groups . . . 41

Index **42**

as_tibble *Convert a testflow object to a one-row tibble*

Description

Convert a testflow object to a one-row tibble

Usage

```
as_tibble(x, ...)
```

Arguments

x	A testflow object.
...	Unused.

Value

A one-row tibble with the workflow name, design, variables, recommended test, null hypothesis, statistic, degrees of freedom when available, p-value, confidence interval when available, effect-size fields, and decision text.

make_cardio_data *Simulate a small cardiovascular teaching dataset*

Description

Simulate a small cardiovascular teaching dataset

Usage

```
make_cardio_data(n = 180, seed = 2026)
```

Arguments

n	Number of rows.
seed	Random seed.

Value

A tibble with example numeric and categorical variables.

plot.testflow *Plot a testflow object*

Description

Plot a testflow object

Usage

```
## S3 method for class 'testflow'
plot(x, title = NULL, subtitle = NULL, caption = NULL, ...)
```

Arguments

x	A testflow object.
title	Optional plot title override. Defaults to the stored title.
subtitle	Optional plot subtitle override. Defaults to the stored subtitle.
caption	Optional plot caption override. Defaults to the stored caption.
...	Unused.

Value

A ggplot object stored in the testflow object, with optional title, subtitle, and caption overrides applied. If the workflow was created with plot = FALSE, returns NULL.

print.summary.sample_size *Print a summary for a sample size object*

Description

Print a summary for a sample size object

Usage

```
## S3 method for class 'summary.sample_size'
print(x, ...)
```

Arguments

x	A summary.sample_size object.
...	Unused.

Value

The object invisibly.

print.testflow *Print a testflow object*

Description

Print a testflow object.

Usage

```
## S3 method for class 'testflow'
print(x, ...)
```

Arguments

x	A testflow object.
...	Unused.

Details

Console colors are enabled by default in interactive sessions. Use `options(testflow.cli_colors = FALSE)` to disable colors, or `options(testflow.cli_colors = TRUE)` to force colors in non-interactive output.

Value

The input `testflow` object, invisibly. Called for its side effect of printing a formatted workflow summary to the console.

print.testflow_sumtab *Print a testflow summary table*

Description

Prints a `sumtab()` result without column truncation, unlike the default tibble print method, which hides columns and truncates cell text (e.g. `"58.0 (10.4). . . "`) once the table exceeds the console width.

Usage

```
## S3 method for class 'testflow_sumtab'
print(x, ...)
```

Arguments

x	A `testflow_sumtab` object, as returned by `sumtab()`.
...	Unused.

Value

The input object, invisibly.

`report` *Return a ready-to-use testflow report*

Description

Return a ready-to-use testflow report

Usage

```
report(x, ...)
```

Arguments

`x`	A testflow object.
`...`	Unused.

Value

A length-one character vector containing a plain-language summary of the workflow result, including the design, recommended test, p-value, effect size when reported, and null hypothesis when available.

`report_test` *Return a ready-to-use testflow report*

Description

Return a ready-to-use testflow report

Usage

```
report_test(x)
```

Arguments

`x`	A testflow object.

Value

A length-one character vector containing the same report text as `report()` for a `testflow` object.

sample_size *Sample size planning for common trial designs*

Description

Planning functions for common trial designs across continuous, binary, survival, and ordinal endpoints.

Usage

```
sample_size(
  endpoint = c("continuous", "binary", "survival", "ordinal"),
  design = c("parallel", "paired", "repeated"),
  objective = c("superiority", "noninferiority", "equivalence"),
  ...
)

sample_size_continuous(
  design = c("parallel", "paired", "repeated"),
  objective = c("superiority", "noninferiority", "equivalence"),
  delta = NULL,
  sd = NULL,
  sd_diff = NULL,
  expected_difference = 0,
  margin = NULL,
  alpha = 0.05,
  power = 0.9,
  allocation = 1,
  dropout = 0,
  n_time = 2,
  correlation = 0.5
)

sample_size_binary(
  design = c("parallel", "paired", "repeated"),
  objective = c("superiority", "noninferiority", "equivalence"),
  p1,
  p2,
  margin = NULL,
  method = c("pooled", "anticipated"),
  discordant_or = NULL,
  discordance_rate = NULL,
  p10 = NULL,
  p01 = NULL,
  alpha = 0.05,
  power = 0.9,
  allocation = 1,
```

```
  dropout = 0,
  n_time = 2
)

sample_size_survival(
  design = c("parallel"),
  objective = c("superiority", "noninferiority", "equivalence"),
  hr,
  margin_hr = NULL,
  lower = NULL,
  upper = NULL,
  survival_a = NULL,
  survival_b = NULL,
  alpha = 0.05,
  power = 0.9,
  allocation = 1,
  dropout = 0,
  method = c("exponential", "ph_only"),
  accrual_duration = NULL,
  follow_up = NULL
)

sample_size_ordinal(
  design = c("parallel"),
  objective = c("superiority"),
  p_superiority = NULL,
  alpha = 0.05,
  power = 0.9,
  dropout = 0
)

sample_size_adjust_dropout(n, dropout = 0)
```

Arguments

endpoint	Endpoint family: continuous, binary, survival, or ordinal.
design	Study design: parallel, paired, or repeated. Repeated designs are routed to the paired formulas when `n_time = 2`.
objective	Planning objective.
delta	Effect size or margin on the continuous scale.
sd	Between-subject standard deviation.
sd_diff	Paired-difference standard deviation.
expected_difference	Planned difference from the null boundary.
margin	Non-inferiority or equivalence margin on the risk-difference or continuous scale.
method	For `sample_size_binary()`, the parallel superiority variance estimator: `pooled` (null variance) or `anticipated` (alternative-hypothesis variance). For `sample_size_survival()`, the event-count method: `exponential` or `ph_only`.

p1, p2	Anticipated binary response probabilities.
discordant_or	Discordant odds ratio for paired binary planning.
discordance_rate	Probability of a discordant pair.
p10, p01	Alternative paired-binary inputs for the discordant cells.
hr	Hazard ratio.
margin_hr	Non-inferiority hazard-ratio margin.
lower, upper	Equivalence bounds on the hazard-ratio scale.
survival_a, survival_b	Survival probabilities at the planning horizon.
p_superiority	Probability that a randomly selected subject from group A exceeds one from group B.
alpha	Type I error rate.
power	Target power.
allocation	Allocation ratio n_B/n_A.
dropout	Expected dropout proportion.
n_time	Number of repeated measures.
correlation	Within-subject correlation for repeated measures when design = "repeated" and n_time > 2.
n	Evaluable sample size before dropout adjustment.
accrual_duration	Uniform accrual (enrollment) duration for sample_size_survival(). When supplied with follow_up, event probabilities account for staggered enrollment. Requires survival_a/survival_b.
follow_up	Additional follow-up duration after accrual ends, for sample_size_survival(). Total study duration is accrual_duration + follow_up.
...	Endpoint-specific arguments passed to the selected helper.

Details

The implementation currently covers:

- Continuous endpoints: parallel and paired superiority, non-inferiority, and equivalence, plus repeated-measures approximation for 3+ time points.
- Binary endpoints: parallel risk-difference planning, paired discordant-pair planning, non-inferiority, and equivalence.
- Survival endpoints: event-based superiority, non-inferiority, and equivalence.
- Ordinal endpoints: Noether's superiority approximation.

The helper sample_size_adjust_dropout() applies $\lceil n/(1-d) \rceil$.

See the individual function help pages (e.g. ?sample_size_continuous) for the underlying formulas and their literature references.

Value

A `sample_size` object with the raw sample size, dropout-adjusted sample size, method label, assumptions, report text, and a plot-ready summary. Formulas and literature references are documented in the individual function help pages, not returned in the object.

References

Julious SA. *Sample Sizes for Clinical Trials*. Chapman & Hall/CRC; 2010.

Examples

```
sample_size_continuous(
  design = "paired",
  objective = "superiority",
  delta = 5,
  sd_diff = 10,
  alpha = 0.05,
  power = 0.90
)

sample_size_binary(
  design = "paired",
  objective = "superiority",
  p10 = 0.2,
  p01 = 0.1,
  alpha = 0.05,
  power = 0.90
)

sample_size_continuous(
  design = "repeated",
  n_time = 4,
  correlation = 0.5,
  objective = "superiority",
  delta = 5,
  sd_diff = 10,
  alpha = 0.05,
  power = 0.90
)
```

`sample_size_bioequivalence`

Sample size for average bioequivalence

Description

Sample size for average bioequivalence

Usage

```
sample_size_bioequivalence(
  design = c("crossover", "parallel"),
  gmr = 1,
  cv_within = NULL,
  cv_between = NULL,
  lower = 0.8,
  upper = 1.25,
  alpha = 0.05,
  power = 0.9,
  allocation = 1,
  dropout = 0,
  method = c("iterative_tost", "normal_approx")
)
```

Arguments

design	crossover (two-period two-treatment) or parallel.
gmr	Anticipated geometric mean ratio (test/reference).
cv_within	Within-subject coefficient of variation on the raw (untransformed) scale. Required for design = "crossover".
cv_between	Between-subject coefficient of variation on the raw scale. Required for design = "parallel".
lower	Lower bioequivalence bound (ratio scale), usually 0.80.
upper	Upper bioequivalence bound (ratio scale), usually 1.25.
alpha	Type I error rate (one-sided; bioequivalence uses two one-sided tests, each at alpha).
power	Target power.
allocation	Allocation ratio n_B / n_A. Only used for design = "parallel".
dropout	Expected dropout proportion.
method	iterative_tost (search for the smallest n achieving the target two-one-sided-test power, computed exactly via the noncentral t distribution; the preferred method) or normal_approx (closed-form approximation treating the variance as known).

Details

On the log scale, with $\theta_0 = \log(GMR)$, $\theta_L = \log(L)$, $\theta_U = \log(U)$, and $d_{BE} = \min(\theta_0 - \theta_L,\ \theta_U - \theta_0)$ (must be positive: the anticipated GMR must lie strictly inside the bioequivalence bounds):

method = "iterative_tost" searches for the smallest n such that the exact two-one-sided-test power is at least the target. Let $SE(n) = \sqrt{2\sigma_w^2/n}$ for a crossover design (total n, $df = n - 2$) or $SE(n_A) = \sigma_b\sqrt{(r+1)/(rn_A)}$ for a parallel design (per-arm n_A, $n_B = rn_A$, $df = n_A + n_B - 2$). The two one-sided test statistics $T_U = (\hat{\theta}_0 - \theta_U)/SE$ and $T_L = (\hat{\theta}_0 - \theta_L)/SE$ each follow a *noncentral* t distribution with the above df and noncentrality parameters $\delta_U = (\theta_0 - \theta_U)/SE$, $\delta_L = (\theta_0 - \theta_L)/SE$

- not a location-shifted central t, and not a normal approximation, since the estimated SE carries its own sampling variability. Power is

$$Power(n) = P(T_U < -t_{1-\alpha,df}) + P(T_L > t_{1-\alpha,df}) - 1 = T_{df,\delta_U}(-t_{1-\alpha,df}) - T_{df,\delta_L}(t_{1-\alpha,df})$$

where $T_{df,\delta}$ is the noncentral t CDF (`stats::pt(..., ncp = delta)`) - this is Phillips's (1990) formula, verified in package tests to match `PowerTOST::power.TOST()` (the regulatory-standard, Owen's-Q-based calculation) to numerical precision for balanced designs. Coefficients of variation are converted to log-scale standard deviations via $\sigma = \sqrt{\log(1 + CV^2)}$.

`method = "normal_approx"` uses the closed-form approximation

$$n_{total} \approx \frac{2\sigma_w^2(z_{1-\beta} + z_{1-\alpha})^2}{d_{BE}^2}$$

(crossover) or, per arm,

$$n_A \approx \frac{(r+1)\sigma_b^2(z_{1-\beta} + z_{1-\alpha})^2}{r d_{BE}^2}$$

(parallel), switching to $z_{1-\beta/2}$ in place of $z_{1-\beta}$ when $GMR = 1$ exactly, matching the analogous special case in `sample_size_continuous()`. `iterative_tost` is preferred because this approximation - which treats σ as known and uses normal rather than noncentral-t quantiles - can under-size a study by a material margin (verified up to ~20% at small n) away from $GMR = 1$ or at small n.

Value

A `sample_size` object.

References

Julious SA. *Sample Sizes for Clinical Trials*. Chapman & Hall/CRC; 2010.

Phillips KF. Power of the two one-sided tests procedure in bioequivalence. *Journal of Pharmacokinetics and Biopharmaceutics*. 1990;18(2):137-144.

`sample_size_cluster_adjust` *Adjust an individually randomized sample size for cluster randomization*

Description

Adjust an individually randomized sample size for cluster randomization

Usage

```
sample_size_cluster_adjust(n_ind, m, rho, cv_m = NULL)
```

Arguments

n_ind	Individually randomized sample size (e.g. from sample_size_continuous() or sample_size_binary()).
m	Average cluster size.
rho	Intracluster correlation.
cv_m	Coefficient of variation of cluster sizes, for unequal cluster sizes. NULL (the default) assumes equal cluster sizes.

Details

Equal cluster size:

$$DE = 1 + (m-1)\rho, \qquad n_{clustered} = \lceil n_{ind} \cdot DE \rceil$$

Unequal cluster size (approximate), with coefficient of variation CV_m:

$$DE \approx 1 + \left((1 + CV_m^2)m - 1\right)\rho$$

Value

The cluster-adjusted sample size.

References

Donner A, Klar N. *Design and Analysis of Cluster Randomization Trials in Health Research*. Arnold; 2000.

---

sample_size_precision *Sample size for confidence-interval precision*

Description

Sample size for confidence-interval precision

Usage

```
sample_size_precision(
  endpoint = c("continuous", "binary"),
  design = c("one_sample", "two_sample", "odds_ratio"),
  width,
  sd = NULL,
  p = NULL,
  p1 = NULL,
  p2 = NULL,
  alpha = 0.05,
  allocation = 1,
```

```
  dropout = 0,
  conservative = FALSE,
  method = c("wald", "wilson", "exact"),
  criterion = c("expected", "worst_case"),
  min_expected_events = 5,
  max_n = 1e+07
)
```

Arguments

endpoint	Endpoint family: continuous or binary.
design	one_sample, two_sample, or (binary only) odds_ratio.
width	Desired confidence-interval half-width, on the scale of the estimate (or of the log odds ratio for design = "odds_ratio").
sd	Standard deviation (continuous endpoint).
p	Anticipated proportion (binary, one_sample).
p1	Anticipated proportion in arm A (binary, two_sample or odds_ratio).
p2	Anticipated proportion in arm B (binary, two_sample or odds_ratio).
alpha	Type I error rate (two-sided CI coverage is 1 - alpha).
allocation	Allocation ratio n_B / n_A. Used for endpoint = "continuous", design = "two_sample"; binary design = "two_sample" (risk-difference half-width, unequal-allocation variance); and design = "odds_ratio".
dropout	Expected dropout proportion.
conservative	For endpoint = "binary", design = "one_sample" only: use the worst-case p = 0.5 instead of the anticipated p. Only supported with method = "wald".
method	For endpoint = "binary", design = "one_sample" only: the confidence-interval method used for planning. "wald" (default, backward-compatible) uses the closed-form normal-approximation formula. "wilson" and "exact" (Clopper-Pearson) instead run a deterministic integer search (see Details) for the minimum sample size whose confidence interval has a half-width no greater than width.
criterion	For endpoint = "binary", design = "one_sample", method %in% c("wilson", "exact") only: "expected" (default) evaluates precision at the single anticipated event count round(n * p). "worst_case" requires the precision target to hold over a prevalence-local band of plausible event counts (see Details). Recorded but does not alter the "wald" closed-form calculation.
min_expected_events	For endpoint = "binary", design = "one_sample" only: threshold below which the expected event count or expected non-event count triggers a rare-event diagnostic warning.
max_n	For endpoint = "binary", design = "one_sample", method %in% c("wilson", "exact") only: upper bound for the minimum-n search. The search errors informatively if no n <= max_n satisfies the requested precision.

Details

Unlike the other `sample_size_*()` functions, precision-based planning is driven by a target confidence-interval half-width w rather than power against an effect size.

Continuous, one sample:

$$n = \left(\frac{z_{1-\alpha/2}\sigma}{w}\right)^2$$

Continuous, two independent means:

$$n_A = \frac{(r+1)\sigma^2 z_{1-\alpha/2}^2}{rw^2}$$

Binary, one proportion, `method = "wald"` (normal approximation; the original, backward-compatible formula):

$$n = \frac{z_{1-\alpha/2}^2 p(1-p)}{w^2}$$

(or $n = z_{1-\alpha/2}^2/(4w^2)$ for the conservative $p = 0.5$ case)

Binary, one proportion, `method = "wilson"` or `"exact"`: there is no closed form; n is found by a deterministic integer search. The planning criterion is the maximum one-sided half-width $\max(\hat{p} - lower,\ upper - \hat{p}) \leq w$; because Wilson and Clopper-Pearson intervals are asymmetric around $\hat{p}$, this is not half of the total interval width. The interval is evaluated at:

- `method = "wilson"`, `criterion = "expected"`: the anticipated proportion itself, $\hat{p} = p$, with no rounding of the event count. The half-width then decreases steadily with n, and the result is the smallest n with

$$\frac{z}{1+z^2/n}\sqrt{\frac{p(1-p)}{n} + \frac{z^2}{4n^2}} + \left|\frac{p + z^2/(2n)}{1+z^2/n} - p\right| \leq w,$$

  which can be checked by hand.

- `method = "exact"`, `criterion = "expected"`: the anticipated integer count $x = \mathrm{round}(np)$ (Clopper-Pearson limits are defined for integer counts).

- `criterion = "worst_case"`: every integer count in a prevalence-local band `qbinom(c(0.001, 0.999), n, p)`.

With integer counts the half-width is not monotone in n: it can meet the target at some n and miss it again at a larger n. The function then returns the smallest n from which the target is met for every larger n (checked over a forward window of $\max(50, 4/p)$ values, and verified against a brute-force scan in the package tests), so that recruiting a few more participants never breaks the planned precision. A Clopper-Pearson interval guarantees at-least-nominal coverage at the evaluated count(s); this is not a guarantee on the achieved width at every conceivable outcome.

A precision-based sample size answers "how wide will my confidence interval be", not "how likely am I to observe any events at all"; for rare-prevalence planning (e.g. newborn-screening or rare-adverse-event studies) the expected event count, and the probability of observing zero events, are reported as diagnostics precisely because a narrow interval does not guarantee that any events will be observed.

Binary, two independent proportions, risk-difference half-width, with allocation ratio $r = n_B/n_A$:

$$n_A = \frac{z^2_{1-\alpha/2}}{w^2}\left[p_A(1-p_A) + p_B(1-p_B)/r\right], \qquad n_B = r n_A$$

(reduces to the earlier equal-allocation-only formula when $r = 1$).

Binary, log-odds-ratio half-width:

$$n_A = \frac{z^2_{1-\alpha/2}}{w^2}\left[\frac{1}{p_A} + \frac{1}{1-p_A} + \frac{1}{r p_B} + \frac{1}{r(1-p_B)}\right]$$

Value

A `sample_size` object. For one-sample binary precision planning, `x$diagnostics` additionally holds rare-event and achieved-precision diagnostics (see Details).

References

Julious SA. *Sample Sizes for Clinical Trials*. Chapman & Hall/CRC; 2010.

Wilson EB. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*. 1927;22(158):209-212.

Clopper CJ, Pearson ES. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*. 1934;26(4):404-413.

Examples

```
# 1) Common prevalence, Wald (backward-compatible default)
sample_size_precision(endpoint = "binary", design = "one_sample", width = 0.05, p = 0.30)

# 2) Rare prevalence, Wilson score, iterative search
sample_size_precision(
  endpoint = "binary", design = "one_sample", width = 0.02,
  p = 0.01, method = "wilson"
)

# 3) Rare prevalence, exact Clopper-Pearson, worst-case criterion
sample_size_precision(
  endpoint = "binary", design = "one_sample", width = 0.005,
  p = 0.001, method = "exact", criterion = "worst_case"
)

# 4) Dropout inflation: complete_case_n vs. adjusted_n
x <- sample_size_precision(
  endpoint = "binary", design = "one_sample", width = 0.02,
  p = 0.05, method = "wilson", dropout = 0.10
)
x$diagnostics$complete_case_n
x$diagnostics$adjusted_n

# 5) Two independent proportions, unequal allocation
sample_size_precision(
```

```
  endpoint = "binary", design = "two_sample", width = 0.08,
  p1 = 0.4, p2 = 0.3, allocation = 2
)
```

`summary.testflow` *Summarize a testflow object*

Description

Summarize a testflow object.

Usage

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

Arguments

`object`	A testflow object.
`...`	Unused.

Details

Console colors follow the same `testflow.cli_colors` option used by `print.testflow()`.

Value

A `summary.testflow` list containing the workflow metadata, descriptives, assumptions, recommended test, primary and alternative test results, post-hoc results when available, effect size, decision, and report text.

`sumtab` *Build a compact descriptive summary table*

Description

Builds a compact table of descriptive summaries using a formula interface. Numeric variables are summarized as mean (SD); median [Q1, Q3]; n. Categorical variables are summarized as n (percent).

Usage

```
sumtab(
  formula,
  data,
  p_value = FALSE,
  overall = TRUE,
  digits = 1,
  p_digits = 3,
  alpha = 0.05,
  fisher_threshold = 5,
  na.rm = TRUE
)
```

Arguments

formula	A one-sided formula such as ~ age + sex or ~ age + sex | treatment.
data	A data frame.
p_value	Logical; add a p-value column when a grouping variable is supplied.
overall	Logical; include an overall summary column.
digits	Number of digits for summary statistics.
p_digits	Number of digits for formatted p-values.
alpha	Significance level used by automatic test selection.
fisher_threshold	Expected-count threshold for Fisher's exact test.
na.rm	Logical; remove missing values before summaries and tests.

Details

When p_value = TRUE and a grouping variable is supplied, sumtab() chooses the p-value test automatically. Numeric variables use Student t-test, Welch t-test, or Wilcoxon rank-sum test for two groups, and one-way ANOVA, Welch ANOVA, or Kruskal-Wallis test for more than two groups. Categorical variables use a chi-square test unless expected counts fall below fisher_threshold, in which case Fisher's exact test is used.

Value

A tibble with one row per numeric variable and one row per categorical level.

Examples

```
dat <- make_cardio_data(80, seed = 1)
sumtab(~ age + sex | treatment, dat, p_value = TRUE)
```

`test_agreement` *Inter-rater agreement workflow (Cohen's kappa)*

Description

Inter-rater agreement workflow (Cohen's kappa)

Usage

```
test_agreement(data, rater1, rater2, alpha = 0.05, plot = TRUE, na.rm = TRUE)
```

Arguments

`data`	A data frame.
`rater1`	First rater's categorical column.
`rater2`	Second rater's categorical column, using the same category set as `rater1`.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Details

With observed agreement p_o (the proportion of subjects on which both raters agree) and chance-expected agreement p_e (from the marginal category proportions):

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

Two different standard errors are used for two different purposes, since the sampling variance of $\hat{\kappa}$ depends on κ itself:

- The confidence interval uses the large-sample SE of Fleiss, Cohen & Everitt (1969), evaluated *at* $\hat{\kappa}$, which accounts for the full marginal covariance structure (not the simpler $\sqrt{p_o(1-p_o)}/[(1-p_e)\sqrt{n}]$ approximation sometimes seen, which understates the variance when categories are unevenly distributed).
- The z-test of $H_0 : \kappa = 0$ uses a *different*, smaller-magnitude SE (Fleiss, 1981), evaluated under the null $\kappa = 0$ rather than at $\hat{\kappa}$: $SE_0 = \sqrt{p_e + p_e^2 - \sum_i \pi_{i\cdot}\pi_{\cdot i}(\pi_{i\cdot} + \pi_{\cdot i})}/[(1-p_e)\sqrt{n}]$. Using the confidence-interval SE for this test instead (a mistake this function previously made) understates the null SE whenever agreement is materially above chance - the typical case - inflating the z-statistic and anti-conservatively shrinking the p-value; verified in package tests to match `irr::kappa2()`'s reported z exactly.

Value

A `testflow` object with class `testflow_agreement`. The object is a list containing the cleaned data, categorical descriptives, the test of kappa against 0 as the primary result with null hypothesis, the agreement (confusion) table, Cohen's kappa as the effect size, optional `ggplot`, original call, and report text.

References

Cohen J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*. 1960;20(1):37-46.

Fleiss JL, Cohen J, Everitt BS. Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin*. 1969;72(5):323-327.

Fleiss JL. *Statistical Methods for Rates and Proportions* (2nd ed.). Wiley; 1981. (Null-hypothesis SE for testing kappa = 0.)

Landis JR, Koch GG. The measurement of observer agreement for categorical data. *Biometrics*. 1977;33(1):159-174. (Magnitude convention.)

`test_categorical` *Test association between two categorical variables*

Description

Test association between two categorical variables

Usage

```
test_categorical(
  formula,
  data,
  y = NULL,
  alpha = 0.05,
  fisher_threshold = 5,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`formula`	A formula such as `x ~ y`, or a data frame when using pipe/data-first style.
`data`	A data frame, or a first categorical column when using data-first style.
`y`	Second categorical column. Optional when using formula style.
`alpha`	Significance level.
`fisher_threshold`	Expected-count threshold for Fisher's exact test.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Value

A `testflow` object with class `testflow_categorical`. The object is a list containing the cleaned data, categorical descriptives, assumption checks, recommended association test, primary test result with null hypothesis, alternative chi-square and Fisher results, effect size, optional `ggplot`, original call, and report text.

References

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, 50(302), 157-175.

Fisher, R. A. (1922). On the interpretation of chi-square from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society*, 85(1), 87-94.

Cramer, H. (1946). *Mathematical Methods of Statistics*. Princeton.

`test_correlation` *Test correlation between two numeric variables*

Description

Test correlation between two numeric variables

Usage

```
test_correlation(
  formula,
  data,
  y = NULL,
  method = c("auto", "pearson", "spearman", "kendall"),
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`formula`	A formula such as `y ~ x`, or a data frame when using pipe/data-first style.
`data`	A data frame, or a first numeric column when using data-first style.
`y`	Second numeric column. Optional when using formula style.
`method`	Correlation method or `"auto"`.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Value

A `testflow` object with class `testflow_correlation`. The object is a list containing the cleaned complete-case data, numeric descriptives, assumption checks, recommended correlation method, primary correlation test with null hypothesis, Pearson/Spearman/Kendall results, a correlation table, effect size, optional `ggplot`, original call, and report text.

References

Pearson, K. (1895). Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58, 240-242.

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1), 72-101.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2), 81-93.

`test_correlation_matrix` *Test a correlation matrix*

Description

Test a correlation matrix

Usage

```
test_correlation_matrix(
  data,
  vars,
  method = c("spearman", "pearson", "kendall"),
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`data`	A data frame.
`vars`	Numeric columns.
`method`	Correlation method.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Value

A `testflow` object with class `testflow_correlation_matrix`. The object is a list containing the cleaned data, numeric descriptives, screening assumptions, selected correlation-matrix method, pairwise correlation matrix, pairwise p-value table, maximum absolute correlation as an effect-size summary, optional heatmap `ggplot`, original call, and report text.

`test_cox` *Cox proportional hazards regression workflow*

Description

Cox proportional hazards regression workflow

Usage

```
test_cox(formula, data, alpha = 0.05, plot = TRUE, na.rm = TRUE)
```

Arguments

`formula`	A formula such as `Surv(time, status) ~ x1 + x2`.
`data`	A data frame.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Details

With predictors $x_1, \ldots, x_p$, the Cox model leaves the baseline hazard $h_0(t)$ unspecified:

$$h(t \mid x) = h_0(t) \exp(\beta_1 x_1 + \cdots + \beta_p x_p)$$

Coefficients are reported as hazard ratios $HR_j = e^{\beta_j}$. The overall model test is the likelihood-ratio test against the model with no predictors, on p degrees of freedom (or more, for multi-level factor terms).

The proportional-hazards assumption is checked via the correlation between scaled Schoenfeld residuals and time (Grambsch & Therneau, 1994); a significant test suggests a predictor's effect on the hazard changes over time.

Value

A `testflow` object with class `testflow_cox`. The object is a list containing the cleaned data, descriptive statistics, a proportional- hazards assumption check, the overall likelihood-ratio test as the primary result with null hypothesis, the per-term hazard-ratio coefficient table, the concordance index as the effect size, optional `ggplot`, original call, and report text.

References

Cox DR. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B*. 1972;34(2):187-202.

Grambsch PM, Therneau TM. Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*. 1994;81(3):515-526.

`test_diagnostic` *Diagnostic test accuracy workflow*

Description

Diagnostic test accuracy workflow

Usage

```
test_diagnostic(data, test, reference, alpha = 0.05, plot = TRUE, na.rm = TRUE)
```

Arguments

`data`	A data frame.
`test`	Binary test-result column (two levels; the alphabetically second level is treated as "positive").
`reference`	Binary gold-standard column (two levels; the alphabetically second level is treated as "positive").
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Details

With true positives TP, false positives FP, false negatives FN, and true negatives TN:

$$Sensitivity = \frac{TP}{TP + FN}, \qquad Specificity = \frac{TN}{TN + FP}$$

$$PPV = \frac{TP}{TP + FP}, \qquad NPV = \frac{TN}{TN + FN}$$

$$LR^{+} = \frac{Sensitivity}{1 - Specificity}, \qquad LR^{-} = \frac{1 - Sensitivity}{Specificity}$$

Sensitivity, specificity, PPV, NPV, and accuracy each get an exact (Clopper-Pearson) confidence interval. Likelihood ratios get the closed-form log-scale interval of Simel, Samsa & Matchar (1991):

$$SE[\log LR^{+}] = \sqrt{\frac{1}{TP} - \frac{1}{TP + FN} + \frac{1}{FP} - \frac{1}{FP + TN}}$$

The primary test compares overall accuracy to the no-information rate (the larger of the two reference-class proportions) via an exact binomial test, following the same convention as `caret::confusionMatrix()`.

Value

A `testflow` object with class `testflow_diagnostic`. The object is a list containing the cleaned data, categorical descriptives, the accuracy-vs-no-information-rate test as the primary result with null hypothesis, a table of sensitivity/specificity/predictive values/ likelihood ratios with confidence intervals, the confusion matrix, accuracy as the effect size, optional `ggplot`, original call, and report text.

References

Simel DL, Samsa GP, Matchar DB. Likelihood ratios with confidence: sample size estimation for diagnostic test studies. *Journal of Clinical Epidemiology*. 1991;44(8):763-770.

Altman DG, Bland JM. Diagnostic tests 1: sensitivity and specificity. *BMJ*. 1994;308(6943):1552.

`test_factorial` *Run a factorial ANOVA workflow*

Description

Run a factorial ANOVA workflow

Usage

```
test_factorial(
  formula,
  data,
  factors = NULL,
  alpha = 0.05,
  type = 2,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`formula`	A formula such as `outcome ~ factor1 * factor2`, or a data frame when using data-first style.
`data`	A data frame, or the outcome column when using data-first style.
`factors`	Factor columns selected with tidyselect syntax. Optional when using formula style.
`alpha`	Significance level.
`type`	Sums-of-squares type: `1` (sequential, base `aov()`), `2`, or `3` (via `car::Anova()`, required for `type = 2/3`). For unbalanced designs these can give materially different p-values; `type = 2` is a reasonable default when there is no strong prior reason to test factors in a particular order. `type = 3` requires sum-to-zero contrasts, which this function sets automatically.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Value

A `testflow` object with class `testflow_factorial`. The object is a list containing the cleaned data, descriptive statistics, residual and variance assumption checks, recommended factorial ANOVA, primary ANOVA term result with null hypothesis, ANOVA table, effect size, optional `ggplot`, original call, and report text.

References

Fisher, R. A. (1925). *Statistical Methods for Research Workers*. Oliver and Boyd.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Lawrence Erlbaum.

`test_groups` *Compare a numeric outcome across more than two groups*

Description

Compare a numeric outcome across more than two groups

Usage

```
test_groups(
  formula,
  data,
  group = NULL,
  alpha = 0.05,
  posthoc = TRUE,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`formula`	A formula such as `outcome ~ group`, or a data frame when using pipe/data-first style.
`data`	A data frame, or an outcome column when using data-first style.
`group`	Grouping column. Optional when using formula style.
`alpha`	Significance level.
`posthoc`	Logical; compute post-hoc comparisons.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Value

A `testflow` object with class `testflow_groups`. The object is a list containing the cleaned data, grouped descriptive statistics, assumption checks, recommended omnibus test, primary test result with null hypothesis, alternative omnibus results, post-hoc comparisons when requested, effect size, optional `ggplot`, original call, and report text.

`test_icc` *Intraclass correlation coefficient workflow*

Description

Intraclass correlation coefficient workflow

Usage

```
test_icc(data, measures, alpha = 0.05, plot = TRUE, na.rm = TRUE)
```

Arguments

`data`	A data frame in wide format (one column per rater/measurement, one row per subject).
`measures`	Rater/measurement columns selected with tidyselect syntax.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Details

Following Shrout & Fleiss (1979), variance components come from one-way (BMS, WMS) and two-way (BMS, JMS, EMS) ANOVA decompositions of the n subjects by k raters:

$$ICC(1,1) = \frac{BMS - WMS}{BMS + (k-1)WMS}$$

$$ICC(2,1) = \frac{BMS - EMS}{BMS + (k-1)EMS + \frac{k}{n}(JMS - EMS)} \quad \text{(two-way random, absolute agreement)}$$

$$ICC(3,1) = \frac{BMS - EMS}{BMS + (k-1)EMS} \quad \text{(two-way fixed, consistency)}$$

ICC(2,1) (absolute agreement, allowing raters to differ systematically) is reported as the primary/effect-size estimate, following the single-measurement, absolute-agreement, two-way random-effects recommendation of Koo & Li (2016) for typical reliability studies. Confidence intervals use the F-distribution-based formulas of McGraw & Wong (1996); ICC(2,1)'s interval uses a Satterthwaite-approximated denominator degrees of freedom.

Value

A `testflow` object with class `testflow_icc`. The object is a list containing the cleaned long-format data, descriptive statistics, the F-test for ICC(2,1) as the primary result with null hypothesis, a table of the one-way (ICC1), two-way random/absolute-agreement (ICC2), and two-way fixed/consistency (ICC3) estimates with confidence intervals, ICC(2,1) as the effect size, optional `ggplot`, original call, and report text.

References

Shrout PE, Fleiss JL. Intraclass correlations: uses in assessing rater reliability. *Psychological Bulletin*. 1979;86(2):420-428.

McGraw KO, Wong SP. Forming inferences about some intraclass correlation coefficients. *Psychological Methods*. 1996;1(1):30-46.

Koo TK, Li MY. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*. 2016;15(2):155-163.

`test_linear_regression` *Multiple linear regression workflow*

Description

Multiple linear regression workflow

Usage

```
test_linear_regression(formula, data, alpha = 0.05, plot = TRUE, na.rm = TRUE)
```

Arguments

`formula`	A formula such as `outcome ~ x1 + x2 + x3`.
`data`	A data frame.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Details

For predictors $x_1, \ldots, x_p$ and n observations, ordinary least squares fits:

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} + \varepsilon_i$$

The overall model test compares the fitted model to an intercept-only model:

$$F = \frac{(SS_{total} - SS_{res})/p}{SS_{res}/(n-p-1)} \sim F_{p,\, n-p-1}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{total}}, \qquad R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

Value

A `testflow` object with class `testflow_linear_regression`. The object is a list containing the cleaned data, descriptive statistics, residual/homoscedasticity/multicollinearity assumption checks, the overall model F-test as the primary result with null hypothesis, the per-term coefficient table, R squared and adjusted R squared as the effect size, optional `ggplot`, original call, and report text.

References

Draper NR, Smith H. *Applied Regression Analysis* (3rd ed.). Wiley; 1998.

`test_logistic_regression`
Logistic regression workflow

Description

Logistic regression workflow

Usage

```
test_logistic_regression(
  formula,
  data,
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`formula`	A formula such as `outcome ~ x1 + x2 + x3`, with a binary (two-level factor or 0/1 numeric) outcome.
`data`	A data frame.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Details

With outcome $y_i \in \{0, 1\}$ and predictors $x_1, \ldots, x_p$:

$$\log \frac{P(y_i = 1)}{1 - P(y_i = 1)} = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi}$$

Coefficients are reported on the log-odds scale and, exponentiated, as odds ratios $OR_j = e^{\beta_j}$.

The overall model test is the likelihood-ratio test against the intercept-only model, using each model's residual deviance $D = -2 \log L$:

$$LR = D_{null} - D_{model} \sim \chi^2_p$$

McFadden's pseudo R squared:

$$R^2_{McFadden} = 1 - \frac{\log L_{model}}{\log L_{null}}$$

Value

A `testflow` object with class `testflow_logistic_regression`. The object is a list containing the cleaned data, descriptive statistics, multicollinearity/influential-point assumption checks, the likelihood-ratio test as the primary result with null hypothesis, the per-term coefficient tables on the log-odds and odds-ratio scale, McFadden's pseudo R squared as the effect size, optional `ggplot`, original call, and report text.

References

Hosmer DW, Lemeshow S, Sturdivant RX. *Applied Logistic Regression* (3rd ed.). Wiley; 2013.

`test_multinomial` *Test multinomial goodness of fit*

Description

Test multinomial goodness of fit

Usage

```
test_multinomial(
  data,
  outcome,
  p = NULL,
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`data`	A data frame.
`outcome`	Categorical outcome column.
`p`	Expected probabilities, or `NULL` for equal probabilities.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Value

A `testflow` object with class `testflow_multinomial`. The object is a list containing the cleaned data, categorical descriptives, assumption checks, recommended goodness-of-fit test, primary chi-square result with null hypothesis, BH-adjusted pairwise binomial checks, effect-size summary, optional `ggplot`, original call, and report text.

References

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, 50(302), 157-175.

`test_one_sample` *Test one numeric sample against a reference value*

Description

Test one numeric sample against a reference value

Usage

```
test_one_sample(
  data,
  outcome,
  mu = 0,
  alternative = c("two.sided", "less", "greater"),
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`data`	A data frame.
`outcome`	Numeric outcome column.
`mu`	Reference value.
`alternative`	Alternative hypothesis.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Value

A `testflow` object with class `testflow_one_sample`. The object is a list containing the cleaned data, descriptive statistics, assumption checks, recommended test, primary test result with null hypothesis, alternative test results, effect size, optional `ggplot`, original call, and report text.

References

Gosset, W. S. (1908). The probable error of a mean. *Biometrika*, 6(1), 1-25.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80-83.

`test_outliers` *Detect numeric outliers*

Description

Detect numeric outliers

Usage

```
test_outliers(
  formula,
  data,
  group = NULL,
  method = c("iqr", "mahalanobis", "both"),
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`formula`	Numeric columns to screen for outliers.
`data`	A data frame.
`group`	Optional grouping column.
`method`	Outlier method.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Value

A `testflow` object with class `testflow_outliers`. The object is a list containing the cleaned data, numeric descriptives, screening assumptions, selected outlier-screening method, flagged IQR and/or Mahalanobis rows, outlier-count summary, optional `ggplot`, original call, and report text. This is a screening workflow, not a single hypothesis test.

`test_paired` *Compare paired before and after numeric measurements*

Description

Compare paired before and after numeric measurements

Usage

```
test_paired(
  formula,
  data,
  after = NULL,
  alternative = c("two.sided", "less", "greater"),
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`formula`	A formula such as `after ~ before`, or a data frame when using data-first style.
`data`	A data frame, or the before column when using data-first style.
`after`	After column. Optional when using formula style.
`alternative`	Alternative hypothesis.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Value

A `testflow` object with class `testflow_paired`. The object is a list containing the cleaned paired data, paired difference, descriptive statistics, assumption checks, recommended paired test, primary test result with null hypothesis, alternative test results, effect size, optional `ggplot`, original call, and report text.

test_paired_categorical *Test paired categorical measurements*

Description

Test paired categorical measurements

Usage

```
test_paired_categorical(
  data,
  before,
  after,
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

data	A data frame.
before	Before categorical column.
after	After categorical column.
alpha	Significance level.
plot	Logical; include a ggplot object.
na.rm	Logical; remove missing values.

Value

A testflow object with class testflow_paired_categorical. The object is a list containing the cleaned paired categorical data, categorical descriptives, assumption checks, McNemar test result, discordant-pair table, optional ggplot, original call, and report text.

test_proportion *Test a one-sample proportion*

Description

Test a one-sample proportion

Usage

```
test_proportion(
  data,
  outcome,
  success,
  p = 0.5,
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

data	A data frame.
outcome	Categorical outcome column.
success	Value counted as success.
p	Reference probability.
alpha	Significance level.
plot	Logical; include a ggplot object.
na.rm	Logical; remove missing values.

Value

A testflow object with class testflow_proportion. The object is a list containing the cleaned data, categorical descriptives, assumption checks, recommended exact or approximate one-sample proportion test, primary test result with null hypothesis, alternative exact and approximate results, observed proportion summary, optional ggplot, original call, and report text.

References

Clopper, C. J., & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4), 404-413.

---

test_repeated *Run a repeated-measures workflow from wide data*

Description

Run a repeated-measures workflow from wide data

Usage

```
test_repeated(
  data,
  measures,
  id = NULL,
  between = NULL,
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

data	A data frame.
measures	Repeated numeric columns selected with tidyselect syntax.
id	Optional subject identifier.
between	Optional between-subject factor.
alpha	Significance level.
plot	Logical; include a ggplot object.
na.rm	Logical; remove missing values.

Value

A testflow object with class testflow_repeated. The object is a list containing long-form repeated-measures data, numeric descriptives, assumption checks, recommended repeated-measures ANOVA or Friedman test, primary test result with null hypothesis, alternative repeated-measures results, post-hoc paired comparisons, effect size, optional ggplot, original call, and report text.

References

Fisher, R. A. (1925). *Statistical Methods for Research Workers*. Oliver and Boyd.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200), 675-701.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80-83.

Girden, E. R. (1992). *ANOVA: Repeated Measures*. Sage.

---

test_repeated_categorical

Test repeated categorical measurements

Description

Test repeated categorical measurements

Usage

```
test_repeated_categorical(
  data,
  measures,
  id = NULL,
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

data	A data frame.
measures	Repeated binary columns selected with tidyselect syntax.
id	Optional subject identifier.
alpha	Significance level.
plot	Logical; include a ggplot object.
na.rm	Logical; remove missing values.

Value

A testflow object with class testflow_repeated_categorical. The object is a list containing the cleaned repeated categorical data, descriptive counts, assumption checks, Cochran Q test result with null hypothesis, pairwise McNemar post-hoc comparisons, effect size, optional ggplot, original call, and report text.

References

Cochran, W. G. (1950). The comparison of percentages in matched samples. *Biometrika*, 37(3/4), 256-266.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153-157.

---

test_repeated_long *Run a repeated-measures workflow from long data*

Description

Run a repeated-measures workflow from long data

Usage

```
test_repeated_long(
  data,
  outcome,
  within,
  id,
  between = NULL,
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

data	A data frame.
outcome	Numeric outcome column.
within	Within-subject time/condition column.
id	Subject identifier column.
between	Optional between-subject factor.
alpha	Significance level.
plot	Logical; include a ggplot object.
na.rm	Logical; remove missing values.

Value

A `testflow` object with class `testflow_repeated`. The object is a list containing the cleaned long-format data, numeric descriptives, assumption checks, recommended repeated-measures ANOVA or Friedman test, primary test result with null hypothesis, alternative repeated-measures results, post-hoc paired comparisons, effect size, optional `ggplot`, original call, and report text.

---

test_roc *Receiver operating characteristic (ROC) curve workflow*

Description

Receiver operating characteristic (ROC) curve workflow

Usage

```
test_roc(data, predictor, outcome, alpha = 0.05, plot = TRUE, na.rm = TRUE)
```

Arguments

`data`	A data frame.
`predictor`	Numeric predictor/biomarker column; higher values are assumed to indicate the positive class.
`outcome`	Binary outcome column (two levels; the alphabetically second level is treated as the positive class).
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Details

The area under the ROC curve equals the probability that a randomly chosen positive-class observation has a higher predictor value than a randomly chosen negative-class observation, and is computed from the Mann-Whitney U statistic:

$$AUC = \frac{U}{n_1 n_0}$$

The standard error uses the closed-form Hanley & McNeil (1982) formula (an asymptotically equivalent, but not numerically identical, alternative to DeLong's method):

$$SE(AUC) = \sqrt{\frac{AUC(1-AUC) + (n_1-1)(Q_1 - AUC^2) + (n_0-1)(Q_2 - AUC^2)}{n_1 n_0}}$$

$$Q_1 = \frac{AUC}{2 - AUC}, \qquad Q_2 = \frac{2AUC^2}{1 + AUC}$$

The Youden's J statistic identifies the threshold maximizing $J = Sensitivity + Specificity - 1$.

Value

A `testflow` object with class `testflow_roc`. The object is a list containing the cleaned data, descriptive statistics by outcome class, the test of AUC against 0.5 as the primary result with null hypothesis, the full ROC curve and the Youden's-J-optimal threshold, AUC as the effect size, optional `ggplot`, original call, and report text.

References

Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*. 1982;143(1):29-36.

Youden WJ. Index for rating diagnostic tests. *Cancer*. 1950;3(1):32-35.

Hosmer DW, Lemeshow S, Sturdivant RX. *Applied Logistic Regression* (3rd ed.). Wiley; 2013. (AUC magnitude convention, p. 177.)

`test_survival` *Kaplan-Meier and log-rank survival workflow*

Description

Kaplan-Meier and log-rank survival workflow

Usage

```
test_survival(formula, data, alpha = 0.05, plot = TRUE, na.rm = TRUE)
```

Arguments

`formula`	A formula such as `Surv(time, status) ~ group`, with a two-level grouping factor.
`data`	A data frame.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Details

For groups A and B, the log-rank test compares observed to expected events under the null of equal survival distributions:

$$\chi^2 = \sum_{g \in \{A,B\}} \frac{(O_g - E_g)^2}{V_g} \sim \chi^2_1$$

The companion effect size is the hazard ratio from a univariate Cox model on the same grouping factor, $HR = e^{\hat{\beta}}$, reported with its Wald confidence interval; the hazard ratio is not itself part of the log-rank test and can disagree with it under strongly non-proportional hazards.

Value

A `testflow` object with class `testflow_survival`. The object is a list containing the cleaned data, per-group descriptive survival statistics, the log-rank test as the primary result with null hypothesis, the Kaplan-Meier curve data, a companion univariate hazard ratio as the effect size, optional `ggplot`, original call, and report text.

References

Kaplan EL, Meier P. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*. 1958;53(282):457-481.

Mantel N. Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer Chemotherapy Reports*. 1966;50(3):163-170.

`test_two_groups` *Compare a numeric outcome between two independent groups*

Description

Compare a numeric outcome between two independent groups

Usage

```
test_two_groups(
  formula,
  data,
  group = NULL,
  alternative = c("two.sided", "less", "greater"),
  alpha = 0.05,
  plot = TRUE,
  na.rm = TRUE
)
```

Arguments

`formula`	A formula such as `outcome ~ group`, or a data frame when using pipe/data-first style.
`data`	A data frame, or an outcome column when using data-first style.
`group`	Two-level grouping column. Optional when using formula style.
`alternative`	Alternative hypothesis.
`alpha`	Significance level.
`plot`	Logical; include a ggplot object.
`na.rm`	Logical; remove missing values.

Value

A `testflow` object with class `testflow_two_groups`. The object is a list containing the cleaned data, descriptive statistics by group, assumption checks, recommended test, primary test result with null hypothesis, alternative test results, effect size, optional `ggplot`, original call, and report text.

Index

as_tibble, 3
as_tibble.sample_size *(sample_size)*, 7

make_cardio_data, 3

plot.sample_size *(sample_size)*, 7
plot.testflow, 4
print.sample_size *(sample_size)*, 7
print.summary.sample_size, 4
print.testflow, 5
print.testflow(), *17*
print.testflow_sumtab, 5

report, 6
report(), *6*
report.sample_size *(sample_size)*, 7
report_test, 6

sample_size, 7
sample_size_adjust_dropout *(sample_size)*, 7
sample_size_binary *(sample_size)*, 7
sample_size_binary(), *13*
sample_size_bioequivalence, 10
sample_size_cluster_adjust, 12
sample_size_continuous *(sample_size)*, 7
sample_size_continuous(), *12, 13*
sample_size_ordinal *(sample_size)*, 7
sample_size_precision, 13
sample_size_survival *(sample_size)*, 7
summary.sample_size *(sample_size)*, 7
summary.testflow, 17
sumtab, 17
sumtab(), *5*

test_agreement, 19
test_categorical, 20
test_correlation, 21
test_correlation_matrix, 22
test_cox, 23
test_diagnostic, 24
test_factorial, 25
test_groups, 26
test_icc, 27
test_linear_regression, 28
test_logistic_regression, 29
test_multinomial, 30
test_one_sample, 31
test_outliers, 32
test_paired, 33
test_paired_categorical, 34
test_proportion, 34
test_repeated, 35
test_repeated_categorical, 36
test_repeated_long, 37
test_roc, 38
test_survival, 40
test_two_groups, 41