Package: textileFlame (via r-universe)

November 12, 2026

Type Package

Title Analysis and Reporting of Textile Vertical Flame Tests

Version 0.1.0

Description Tools for validating, summarising, comparing and reporting results of vertical flame tests of textiles, such as those obtained with ASTM D6413/D6413M (2022) <doi:10.1520/D6413_D6413M-22> (afterflame time, afterglow time and char length). Measurements are checked on import, observations that are missing or only partly observed (for example specimens consumed over their full length) are kept distinct from ordinary measurements, and treatment comparisons use independently treated batches rather than individual specimens as the unit of replication by default. Acceptance criteria are never assumed; they must be supplied together with their source.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.0.0)

Imports graphics, grDevices, stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.1

NeedsCompilation no

Author Mark Joshua R. Secretario [aut, cre, cph]

Maintainer Mark Joshua R. Secretario <markjoshuasecretario@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-12 12:59:05 UTC

RemoteUrl https://github.com/cran/textileFlame

RemoteRef HEAD

RemoteSha 13f0f897015f8f7b36a8c42e92750620a6a8c0c1

Contents

compare_treatments . . . 2
evaluate_criteria . . . 4
flame_criteria . . . 5
flame_data . . . 6
flame_example . . . 8
flame_metadata . . . 9
flame_report . . . 9
plot_comparison . . . 10
plot_specimens . . . 11
summarize_flame . . . 11
summarize_observations . . . 13
validate_flame_data . . . 13

Index **15**

`compare_treatments` *Compare treatments with a control*

Description

Estimates, for each endpoint and stratum, the difference between every treatment and the control (treated minus control), with a Welch confidence interval and test, and the percentage reduction relative to the control.

Usage

```
compare_treatments(
  x,
  endpoint = NULL,
  control = NULL,
  strata = NULL,
  unit = c("batch", "specimen"),
  conf_level = 0.95,
  censored = c("na", "exclude", "bound")
)
```

Arguments

`x`	A flame_data object.
`endpoint`	Endpoints to compare; `NULL` for all available.
`control`	Name of the reference treatment. Defaults to the control set in `flame_data()`.
`strata`	Strata within which to compare. Defaults to all strata of `x`; must include all strata, optionally in a different order.
`unit`	Unit of replication: `"batch"` or `"specimen"`; see Details.
`conf_level`	Confidence level.
`censored`	How to treat censored observations; see Details.

Details

Unit of replication. With `unit = "batch"` (the default), specimens from the same independently treated batch are first averaged, and the batch means are the replicates. This avoids inflating the replication when, for example, five strips are cut from one treated fabric. If a group has fewer than two batches, the difference is still reported but no interval or test is computed. With `unit = "specimen"`, specimens are treated as independent; the intervals then describe test-to-test variation within the available batches only and a warning is issued. Batch means are not weighted by the number of specimens.

Percentage reduction is `100 * (mean_control - mean_treated) / mean_control` and is only computed when the control mean is positive. Its interval is a Fieller interval for the ratio of the two means (using the Welch degrees of freedom as an approximation), reported only when the control mean is clearly different from zero; the upper limit is capped at 100 because values cannot be negative.

Censored and missing observations. Missing observations are excluded. Censored observations (recorded values are lower bounds; see `flame_data()`) are handled by `censored`:

- `"na"` (default): comparisons involving any censored observation return no estimate (`status = "not_estimable"`), so that censored values are never treated as ordinary measurements;
- `"exclude"`: censored observations are dropped and the comparison is conditional on the measured values only (flagged in `diff_bound` and `note`); this typically favours the group with more censoring;
- `"bound"`: recorded values are used to give a one-sided bound when only one group is censored (for example an untreated control consumed over its full length). If only the control is censored, the true difference is at most the reported one (`diff_bound = "upper"`) and the true percentage reduction is at least the reported one (`pct_bound = "lower"`). No interval or test is given for bounds, and a comparison with censoring in both groups is not estimable.

P-values are not adjusted for multiple comparisons. These are unpaired comparisons of independent groups, not paired or repeated-measures models. Unknown batch identities disable default batch-level intervals and tests.

Value

A data frame of class `flame_comparison` with one row per endpoint, stratum and non-control treatment, and columns `endpoint`, the strata, `treatment`, `unit`, `n_treated`, `n_control` (numbers of analysis units), `mean_treated`, `mean_control`, `difference`, `conf_low`, `conf_high`, `p_value`, `pct_reduction`, `pct_low`, `pct_high`, `n_censored_treated`, `n_censored_control`, `diff_bound`, `pct_bound`, `status` and `note`.

References

Fieller, E. C. (1954). Some Problems in Interval Estimation. Journal of the Royal Statistical Society, Series B, 16(2), 175–185. doi:10.1111/j.25176161.1954.tb00159.x. The implementation uses an approximate Welch quantile for independent means.

Examples

```
fd <- flame_data(flame_example(), control = "Untreated")
compare_treatments(fd, endpoint = "afterflame")
compare_treatments(fd, endpoint = "char_length", censored = "bound")
```

`evaluate_criteria` *Evaluate user-supplied acceptance criteria*

Description

Applies a flame_criteria object to each group of specimens and returns, for every rule, whether it is met, not met, or cannot be decided.

Usage

```
evaluate_criteria(x, criteria, by = NULL)
```

Arguments

`x`	A flame_data object.
`criteria`	A flame_criteria object.
`by`	Grouping columns, as in `summarize_flame()`.

Details

Results are `"pass"`, `"fail"` or `"indeterminate"`.

- For `statistic = "individual"`, a recorded value above the limit is a `"fail"`, even if it is censored (a lower bound) or other specimens are missing. Without such a value, any missing or censored specimen makes the result `"indeterminate"`, because the unobserved or unbounded value could exceed the limit.
- For `statistic = "mean"`, any missing specimen makes the result `"indeterminate"`. Otherwise a mean (a lower bound if censored values are present) above the limit is a `"fail"`; a censored mean at or below the limit is `"indeterminate"`.

`group_result` combines the rules within a group: `"fail"` if any rule fails, otherwise `"indeterminate"` if any is undecided, otherwise `"pass"`. The specimens within a group are not weighted by batch. Whether a rule set is appropriate for your material, and how many specimens it requires, is determined by the source you cite.

Value

A data frame with the grouping columns, `endpoint`, `statistic`, `limit`, `observed`, `n_specimens`, `n_censored`, `n_missing`, `result`, `note` and `group_result`. The criteria source and edition are stored as attributes `criteria_source` and `criteria_edition`.

Examples

```
fd <- flame_data(flame_example(), control = "Untreated")
crit <- flame_criteria(
  data.frame(endpoint = "char_length", statistic = "individual", max = 150),
  source = "Example specification (illustrative values only)"
)
evaluate_criteria(fd, crit)
```

`flame_criteria` *Define acceptance criteria supplied by the user*

Description

Acceptance rules are never assumed by this package. A criteria object records rules that you take from a named specification, product standard or contract, together with that source (and optionally its edition), so that the origin of every pass/fail statement is reported with it.

Usage

```
flame_criteria(
  rules,
  source,
  edition = NA_character_,
  time_unit = c("s", "min"),
  length_unit = c("mm", "cm", "in")
)
```

Arguments

`rules`	A data frame with columns `endpoint` (`"afterflame"`, `"afterglow"` or `"char_length"`), `statistic` (`"individual"` for the largest single specimen, or `"mean"` for the mean of the specimens in a group) and `max` (the largest allowed value). A rule is met when the statistic does not exceed `max`.
`source`	Required single string naming where the rules come from (for example the specification and clause). There is no default.
`edition`	Optional edition or date of the source.
`time_unit`, `length_unit`	Units in which the limits in `rules` are expressed; they are converted to seconds and millimetres.

Value

An object of class `flame_criteria`.

Examples

```
# Illustrative limits only; take real ones from your governing specification.
crit <- flame_criteria(
  data.frame(endpoint = c("afterflame", "char_length"),
             statistic = c("individual", "mean"),
             max = c(2, 150)),
  source = "Example specification (illustrative values only)"
)
crit
```

`flame_data` *Create a validated vertical flame-test data set*

Description

Imports specimen-level results of a vertical flame test (afterflame time, afterglow time and char length), checks them, and returns an object that the other functions in the package accept.

Usage

```
flame_data(
  data,
  specimen = "specimen",
  treatment = "treatment",
  batch = "batch",
  afterflame = "afterflame",
  afterglow = "afterglow",
  char_length = "char_length",
  direction = "direction",
  laundering = "laundering",
  edition = "edition",
  melt = "melt",
  drip = "drip",
  notes = "notes",
  control = NULL,
  time_unit = c("s", "min"),
  length_unit = c("mm", "cm", "in"),
  method_edition = NA_character_,
  status_suffix = "_status"
)
```

Arguments

`data`	A data frame with one row per specimen.
`specimen`, `treatment`, `batch`	Names of the specimen identifier, treatment and batch columns. `batch = NA` declares the batch unknown.
`afterflame`, `afterglow`, `char_length`	Names of the endpoint columns.
`direction`, `laundering`, `edition`, `melt`, `drip`, `notes`	Names of optional columns. `melt` and `drip` may be logical, 0/1 or yes/no.
`control`	Name of the untreated control level of `treatment`, used as the default reference by `compare_treatments()`.
`time_unit`	Unit of the afterflame and afterglow times in `data`: `"s"` or `"min"`.
`length_unit`	Unit of char length in `data`: `"mm"`, `"cm"` or `"in"`.

`method_edition` Optional single string naming the edition of the test method used (for example `"ASTM D6413/D6413M-22"`). Use the `edition` column instead if editions differ between specimens.

`status_suffix` Suffix that identifies status columns.

Details

Every row of `data` is one specimen. Columns are mapped by name; the defaults are the canonical names shown in the usage. Optional columns (`direction`, `laundering`, `edition`, `melt`, `drip`, `notes`, and any of the three endpoints) are used when a column with the default name exists and are ignored otherwise; if you name a column explicitly and it is absent, an error is raised. Set an argument to `NULL` to ignore a column. Columns that are not mapped are dropped.

Batches. `batch` identifies the independently treated batch (for example one plasma-treated piece of fabric) a specimen was cut from. Specimens from the same batch are not independent treatment replicates, and `compare_treatments()` uses batches as its unit of replication by default. Batch labels only need to be unique within a treatment. If the batch is genuinely unknown, use `batch = NA`; every specimen is then treated as its own batch for bookkeeping and a warning is issued. Batch-level inference is disabled when batches are unknown.

Strata. `direction`, `laundering` and `edition` define strata. By default summaries and comparisons are computed within strata so that unlike measurements are not pooled by accident. Comparisons require every stratum; custom summary grouping may deliberately pool strata.

Observation status. Each endpoint can have a companion status column named by appending `status_suffix` to the endpoint column name (for example `char_length_status`). Allowed values are:

- `"measured"`: an ordinary measurement;
- `"missing"`: not recorded (the value must be empty);
- `"censored"`: the true value is at least the recorded value, for example an observation stopped at a known limit. Use this status only when a lower-bound interpretation is scientifically justified; complete consumption is not automatically assigned a censoring status;
- `"not_applicable"`: the endpoint does not apply to this specimen.

If no status column exists, empty values are `"missing"` and all other values are `"measured"`. Missing and censored observations are never turned into zeros or ordinary measurements.

Units. Values are converted on import and stored in seconds and millimetres. `time_unit` and `length_unit` describe the units of the input and apply to the whole data set.

Value

An object of class `flame_data` (a data frame with the canonical columns `specimen`, `treatment`, `batch`, the available strata and endpoints, and one `<endpoint>_status` column per endpoint) with metadata available from `flame_metadata()`.

See Also

`validate_flame_data()` to inspect every check without stopping.

Examples

```
raw <- flame_example()
fd <- flame_data(raw, control = "Untreated",
                 method_edition = "synthetic example")
fd
```

`flame_example` *Synthetic example data*

Description

Returns a small synthetic data set of vertical flame-test results that illustrates the structure expected by `flame_data()`. The numbers were generated for demonstration and do not represent any real material or laboratory.

Usage

```
flame_example()
```

Details

The data contain 60 specimens: an untreated control (one batch), and two treatments (two batches each), tested in two directions (`warp`, `weft`) and at two laundering levels (0 and 25 cycles), with three specimens per batch and stratum. All untreated specimens are recorded as consumed over an assumed full length of 300 mm (`char_length_status = "censored"`), one treated specimen is also censored, and one afterglow time is missing.

Value

A data frame with columns `specimen`, `treatment`, `batch`, `direction`, `laundering`, `afterflame` (s), `afterglow` (s), `char_length` (mm), the three `*_status` columns, `melt`, `drip` and `notes`.

Examples

```
raw <- flame_example()
head(raw)
table(raw$treatment, raw$char_length_status)
```

flame_metadata *Metadata of a flame_data object*

Description

Metadata of a flame_data object

Usage

```
flame_metadata(x)
```

Arguments

x	A flame_data object.

Value

A list with the control level, the endpoints and strata present, the stored and input units, the method edition, the status-column suffix and the validation `issues` recorded at creation.

Examples

```
fd <- flame_data(flame_example(), control = "Untreated")
flame_metadata(fd)$endpoints
```

flame_report *Write a reproducible Markdown report*

Description

Assembles the data description, endpoint summaries and, optionally, treatment comparisons and acceptance-criteria results into a plain Markdown document that records the units, test-method edition, handling of censored observations and the package version. No fixed wording about pass/fail is added: criteria results are labelled with the source you supplied.

Usage

```
flame_report(
  x,
  comparisons = NULL,
  criteria = NULL,
  file = NULL,
  title = "Vertical flame test report",
  date = Sys.Date(),
  digits = 3
)
```

Arguments

x	A flame_data object.
comparisons	A flame_comparison object from compare_treatments(), TRUE to compute one with default settings, or NULL to omit the section.
criteria	A flame_criteria object to evaluate and include, or NULL.
file	Optional path; if given, the report is written there (UTF-8).
title	Report title.
date	Date shown in the report. Fix it to make the output exactly reproducible.
digits	Number of decimal places in tables.

Value

The report as a character vector of lines, invisibly.

Examples

```
fd <- flame_data(flame_example(), control = "Untreated",
                 method_edition = "synthetic example")
txt <- flame_report(fd, comparisons = TRUE, date = as.Date("2026-01-01"))
head(txt, 12)
```

---

plot_comparison *Plot treatment comparisons*

Description

Draws, for each endpoint, the difference between each treatment and the control with its confidence interval. Filled points are exact estimates; crosses are bounds or conditional estimates (see compare_treatments()), with intervals shown when available. Bounds have no interval. Comparisons that could not be estimated are not drawn.

Usage

```
plot_comparison(x)
```

Arguments

x	A flame_comparison object from compare_treatments().

Value

x, invisibly.

Examples

```
fd <- flame_data(flame_example(), control = "Untreated")
cmp <- compare_treatments(fd, endpoint = "afterflame")
plot_comparison(cmp)
```

plot_specimens *Plot specimen-level results*

Description

Draws every specimen as a point, grouped by treatment, with one panel per stratum. Colours distinguish the batches within a treatment, open triangles mark censored observations (the plotted value is a lower bound), and the horizontal bar is the mean of the plotted values. Missing observations are not drawn.

Usage

```
plot_specimens(x, endpoint = NULL, strata = NULL)

## S3 method for class 'flame_data'
plot(x, y, ...)
```

Arguments

x	A flame_data object.
endpoint	A single endpoint; defaults to the first available.
strata	Strata used to split panels; defaults to all strata of x.
y, ...	Not used.

Value

x, invisibly.

Examples

```
fd <- flame_data(flame_example(), control = "Untreated")
plot_specimens(fd, endpoint = "char_length")
```

summarize_flame *Summarise flame-test endpoints by treatment and stratum*

Description

Computes counts and descriptive statistics for each endpoint within each combination of the grouping columns, always reporting how many observations were measured, censored or missing.

Usage

```
summarize_flame(
  x,
  endpoint = NULL,
  by = NULL,
  censored = c("bound", "exclude", "na")
)
```

Arguments

x	A flame_data object.
endpoint	Endpoints to summarise; NULL for all available.
by	Grouping columns: any of "treatment", "batch" and the strata present. The default is "treatment" plus all strata.
censored	How to treat censored values; see Details.

Details

Censored observations (for example specimens consumed over their full length) have recorded values that are only lower bounds. censored controls how they enter the statistics:

- "bound" (default): recorded values are used, so mean, median, min and max are lower bounds when any censored value is present (bound = "lower_bound"); sd is not reported in that case;
- "exclude": only measured values are used (bound = "censored_excluded"). The statistics are then conditional on the specimen not being censored and are biased towards better performance;
- "na": all statistics are NA when any censored value is present (bound = "not_estimable").

Missing observations are always excluded and counted in n_missing.

Value

A data frame with one row per endpoint and group: the grouping columns, endpoint, n_specimens, n_batches, n_measured, n_censored, n_missing, n_not_applicable, mean, sd, median, min, max and bound.

Examples

```
fd <- flame_data(flame_example(), control = "Untreated")
summarize_flame(fd, endpoint = "char_length")
summarize_flame(fd, endpoint = "afterflame", by = c("treatment", "batch"))
```

summarize_observations *Count melting and dripping observations*

Description

Counts, within each group, the specimens flagged as melting and/or dripping. Requires the `melt` and/or `drip` column to have been supplied to `flame_data()`.

Usage

```
summarize_observations(x, by = NULL)
```

Arguments

`x`	A flame_data object.
`by`	Grouping columns: any of `"treatment"`, `"batch"` and the strata present. The default is `"treatment"` plus all strata.

Value

A data frame with the grouping columns, `n_specimens`, and `n_melt` / `n_drip` (number of specimens flagged `TRUE`) and `n_unrecorded_melt` / `n_unrecorded_drip` for the columns available.

Examples

```
fd <- flame_data(flame_example(), control = "Untreated")
summarize_observations(fd)
```

validate_flame_data *Check flame-test data without stopping*

Description

Runs every validation check that `flame_data()` performs and returns all findings instead of stopping at the first error.

Usage

```
validate_flame_data(data, ...)
```

Arguments

`data`	A data frame with one row per specimen.
`...`	Arguments passed to `flame_data()` (column mappings, units, `control`, and so on).

Value

A data frame of class `flame_issues` with columns `severity` (`"error"`, `"warning"` or `"note"`), `check`, `column`, `rows` and `message`. Errors prevent `flame_data()` from creating an object; warnings and notes describe things worth reviewing.

Examples

```
raw <- flame_example()
raw$char_length[3] <- -5
validate_flame_data(raw, control = "Untreated")
```

Index

compare_treatments, 2
compare_treatments(), *6, 7*, *10*

evaluate_criteria, 4

flame_criteria, *4*, 5, *10*
flame_data, *2*, *4*, 6, *9–13*
flame_data(), *2, 3*, *8*, *13, 14*
flame_example, 8
flame_metadata, 9
flame_metadata(), *7*
flame_report, 9

plot.flame_data *(*plot_specimens*)*, 11
plot_comparison, 10
plot_specimens, 11

summarize_flame, 11
summarize_flame(), *4*
summarize_observations, 13

validate_flame_data, 13
validate_flame_data(), *7*