Package 'falsifyr'

August 7, 2026

Title Adversarial Robustness Attacks for Statistical Claims

Version 1.0.0

Description Attacks fitted R model claims by searching for small plausible perturbations that make a target result disappear. The package focuses on claim-level fragility, smallest-kill reporting, and reproducible caveated robustness checks for ordinary fitted model objects. The methods draw on the fragility-index concept of Walsh et al. (2014) <doi:10.1016/j.jclinepi.2013.10.019>, multiverse analysis of Steegen et al. (2016) <doi:10.1177/1745691616658637>, specification-curve analysis of Simonsohn et al. (2020) <doi:10.1038/s41562-020-0912-z>, and robust covariance estimation of Zeileis (2004) <doi:10.18637/jss.v011.i10>.

License MIT + file LICENSE

URL https://github.com/msaule/falsifyr, https://msaule.github.io/falsifyr/

BugReports https://github.com/msaule/falsifyr/issues

Language en-US

Encoding UTF-8

LazyData true

Depends R (>= 4.1)

Imports cli, ggplot2, parallel, rlang, stats, tibble, vctrs

Suggests broom, knitr, lme4, lmtest, rmarkdown, rstudioapi, sandwich, shiny, survival, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Markuss Saule [aut, cre, cph]

Maintainer Markuss Saule <markusstomas@gmail.com>

Repository CRAN

Date/Publication 2026-08-07 20:20:02 UTC

Contents

`attack` *Attack a statistical claim*

Description

Runs a collection of adversarial robustness checks against a fitted model claim. The returned object summarizes whether the claim survives each attack, the smallest perturbation that kills it, and an overall survival score.

Usage

```
attack(
  model,
  term = NULL,
  data = NULL,
  outcome = NULL,
  cluster = NULL,
 profile = c("default", "clinical", "social_science", "prediction", "strict", "fast"),
  attacks = NULL,
  intensity = c("normal", "fast", "deep", "insane"),
  alpha = 0.05,
  alternative = c("two.sided", "less", "greater"),
 kill_rule = c("p_over_alpha", "ci_crosses_zero", "sign_flip", "effect_below_threshold"),
  effect_threshold = NULL,
  seed = 1,
  parallel = FALSE,
  verbose = TRUE
)
```

Arguments

model	A fitted lm, glm, aov, lme4::lmer, lme4::glmer, or survival::coxph model. htest objects return an explicit limited-support result.
term	Character scalar naming the coefficient or test term to attack. If NULL, falsifyr attacks the first non-intercept coefficient.
data	Optional data frame used to refit the model. When omitted, falsifyr attempts to recover the model data.
outcome	Optional character vector of user-supplied placebo outcome names for the placebo attack family.
cluster	Optional character scalar naming a grouping variable for a grouped row-deletion attack. Supply data when the grouping variable is not part of the fitted formula.
profile	Character scalar choosing an attack profile. Profiles tune the default attack-family emphasis when attacks = NULL; profile = "fast" also defaults to intensity = "fast" when intensity is not supplied.
attacks	Character vector of attack families. NULL runs the default families.
intensity	Character scalar controlling attack breadth: "fast", "normal", "deep", or "insane".
alpha	Significance level used by kill rules.
alternative	Character scalar defining the claim direction for coefficient tests: "two.sided", "less", or "greater".
kill_rule	Character scalar defining what kills a claim. Supported rules are "p_over_alpha", "ci_crosses_zero", "sign_flip", and "effect_below_threshold".
effect_threshold	Numeric threshold used by "effect_below_threshold".
seed	Integer seed for deterministic attack runs.
parallel	Logical; if TRUE, independent attack families run on at most two local workers.
verbose	Logical; if TRUE, prints progress messages for expensive "insane" attack runs.

Value

A falsifyr_attack object with the extracted claim, attack leaderboard, smallest kill, survival score, verdict, runtime metadata, and warnings.

Examples

```
fit <- lm(score ~ treatment + age + baseline_score, data = fragile_trial)
result <- attack(fit, term = "treatment", attacks = "row_deletion", intensity = "fast")
result
```

attack_leaderboard *Extract the attack leaderboard*

Description

Returns the ranked attack table from a `falsifyr_attack` object. The first row is the smallest kill when any attack killed the claim.

Usage

```
attack_leaderboard(result)
```

Arguments

`result`	A `falsifyr_attack` object returned by `attack()`.

Value

A tibble of ranked attack results.

Examples

```
fit <- lm(score ~ treatment + age + baseline_score, data = fragile_trial)
result <- attack(fit, term = "treatment", attacks = "row_deletion", intensity = "fast")
attack_leaderboard(result)
```

attack_this_claim *Launch the Attack This Claim RStudio addin*

Description

Opens a Shiny gadget inside RStudio for selecting a supported model object, choosing a target term, running falsifyr attacks, and viewing the generated report. The result is assigned to `falsifyr_last_attack` in the selected environment.

Usage

```
attack_this_claim(envir = parent.frame())
```

Arguments

`envir`	Environment to scan for supported model objects.

Value

Invisibly returns `NULL` when the addin cannot be launched; otherwise launches the gadget for its side effects.

broom-methods *Convert an attack result to a tidy leaderboard*

Description

Optional broom methods expose falsifyr's two headline data products: a row-per-attack leaderboard from `broom::tidy()` and a one-row claim summary from `broom::glance()`.

Usage

```
tidy.falsifyr_attack(x, ...)

glance.falsifyr_attack(x, ...)
```

Arguments

x	A falsifyr_attack object.
...	Additional arguments, currently ignored.

Value

`tidy.falsifyr_attack()` returns the ranked attack tibble without list-column payloads. `glance.falsifyr_attack()` returns a one-row tibble summarizing the claim, verdict, score, and attack counts.

extract_claim *Extract the statistical claim from a model*

Description

Builds the claim card that falsifyr attacks: term, estimate, uncertainty, p-value, confidence interval, and kill-rule metadata.

Usage

```
extract_claim(model, term = NULL, ...)

## Default S3 method:
extract_claim(model, term = NULL, ...)

## S3 method for class 'lm'
extract_claim(
  model,
  term = NULL,
  alpha = 0.05,
  alternative = c("two.sided", "less", "greater"),
```

```
  kill_rule = "p_over_alpha",
  effect_threshold = NULL,
  ...
)

## S3 method for class 'glm'
extract_claim(
  model,
  term = NULL,
  alpha = 0.05,
  alternative = c("two.sided", "less", "greater"),
  kill_rule = "p_over_alpha",
  effect_threshold = NULL,
  ...
)

## S3 method for class 'htest'
extract_claim(
  model,
  term = NULL,
  alpha = 0.05,
  alternative = c("two.sided", "less", "greater"),
  kill_rule = "p_over_alpha",
  effect_threshold = NULL,
  ...
)

## S3 method for class 'anova'
extract_claim(
  model,
  term = NULL,
  alpha = 0.05,
  alternative = c("two.sided", "less", "greater"),
  kill_rule = "p_over_alpha",
  effect_threshold = NULL,
  ...
)

## S3 method for class 'aov'
extract_claim(
  model,
  term = NULL,
  alpha = 0.05,
  alternative = c("two.sided", "less", "greater"),
  kill_rule = "p_over_alpha",
  effect_threshold = NULL,
  ...
)
```

```
## S3 method for class 'merMod'
extract_claim(
  model,
  term = NULL,
  alpha = 0.05,
  alternative = c("two.sided", "less", "greater"),
  kill_rule = "p_over_alpha",
  effect_threshold = NULL,
  ...
)

## S3 method for class 'coxph'
extract_claim(
  model,
  term = NULL,
  alpha = 0.05,
  alternative = c("two.sided", "less", "greater"),
  kill_rule = "p_over_alpha",
  effect_threshold = NULL,
  ...
)
```

Arguments

`model`	A fitted model or hypothesis-test object.
`term`	Character scalar naming the coefficient or test term.
`...`	Additional arguments passed to methods.
`alpha`	Significance level stored on the extracted claim.
`alternative`	Character scalar defining the claim direction for coefficient tests: `"two.sided"`, `"less"`, or `"greater"`.
`kill_rule`	Character scalar naming the kill rule to store on the extracted claim.
`effect_threshold`	Numeric threshold stored on the claim for `"effect_below_threshold"`.

Value

A list describing the extracted claim.

`fragile_trial` *Synthetic fragile trial data*

Description

A small trial-like data set where the treatment claim starts below `p = 0.05` but is sensitive to row deletion, missing-data alternatives, and measurement-error attacks. It includes deterministic missingness in `baseline_score` to exercise missing-data attacks.

Usage

```
fragile_trial
```

Format

A data frame with 80 rows and 4 variables:

score Continuous outcome.

treatment Binary treatment indicator.

age Participant age.

baseline_score Baseline continuous score with some missing values.

Value

A data frame with one row per simulated participant. The columns contain the outcome, treatment assignment, age, and baseline score used to demonstrate a statistically significant but perturbation-sensitive claim.

Source

Simulated data created for the `falsifyr` package.

`is_killed` *Decide whether a perturbed claim is killed*

Description

Applies the selected kill rule to a claim extracted from a perturbed model.

Usage

```
is_killed(
  claim,
  original_claim = NULL,
  alpha = claim$alpha %||% 0.05,
  kill_rule = claim$kill_rule %||% "p_over_alpha",
  effect_threshold = claim$effect_threshold
)
```

Arguments

`claim`	A claim list, typically produced by `extract_claim()`.
`original_claim`	Optional original claim. Required for `"sign_flip"`.
`alpha`	Significance level for `"p_over_alpha"`.
`kill_rule`	Character scalar naming the kill rule.
`effect_threshold`	Numeric threshold for `"effect_below_threshold"`.

Value

TRUE if the claim is killed, otherwise FALSE.

Examples

```
fit <- lm(score ~ treatment + age, data = fragile_trial)
claim <- extract_claim(fit, term = "treatment")
is_killed(claim)
```

---

refit_model *Refit a model on perturbed data*

Description

Refits a supported model class with a replacement data frame and optional formula. Attack families use this generic internally, and it is exported for users who want reproducible perturbation workflows. For lm and glm model frames, evaluated weights and offsets are preserved where possible.

Usage

```
refit_model(model, data, formula = NULL, ...)

## S3 method for class 'lm'
refit_model(model, data, formula = NULL, ...)

## S3 method for class 'glm'
refit_model(model, data, formula = NULL, ...)

## S3 method for class 'aov'
refit_model(model, data, formula = NULL, ...)

## S3 method for class 'merMod'
refit_model(model, data, formula = NULL, ...)

## S3 method for class 'coxph'
refit_model(model, data, formula = NULL, ...)
```

Arguments

model	A fitted model object.
data	A data frame for the refit.
formula	Optional replacement formula. Defaults to the model formula.
...	Additional arguments passed to the model-fitting function.

Value

A refitted model object of the same broad class as model.

Examples

```
fit <- lm(score ~ treatment + age, data = fragile_trial)
refit_model(fit, data = fragile_trial)
```

---

report *Write an HTML attack report*

Description

Creates a standalone static HTML report for a `falsifyr_attack` object.

Usage

```
report(result, file)
```

Arguments

`result`	A `falsifyr_attack` object returned by `attack()`.
`file`	Output HTML file path. This argument is required; `report()` never writes to the working directory by default.

Value

The normalized output path, invisibly.

Examples

```
fit <- lm(score ~ treatment + age, data = fragile_trial)
result <- attack(fit, term = "treatment", attacks = "row_deletion", intensity = "fast")
out <- report(result, file = tempfile(fileext = ".html"))
file.exists(out)
```

---

resilient_trial *Synthetic resilient trial data*

Description

A small trial-like data set with a stronger treatment effect intended to survive simple falsifyr attack families such as row deletion, robust uncertainty checks, and drop-one covariate attacks.

Usage

```
resilient_trial
```

Format

A data frame with 120 rows and 4 variables:

score Continuous outcome.

treatment Binary treatment indicator.

age Participant age.

baseline_score Baseline continuous score.

Value

A data frame with one row per simulated participant. The columns contain the outcome, treatment assignment, age, and baseline score used to demonstrate a strong claim that survives the package's basic attacks.

Source

Simulated data created for the `falsifyr` package.

`score_survival` *Score claim survival*

Description

Computes falsifyr's heuristic 0-100 survival score from an attack leaderboard.

Usage

```
score_survival(attacks, smallest_kill = NULL)
```

Arguments

`attacks`	A data frame of attack results.
`smallest_kill`	Optional row-like object describing the smallest kill.

Value

Integer survival score from 0 to 100.

Examples

```
fit <- lm(score ~ treatment + age, data = fragile_trial)
result <- attack(fit, term = "treatment", attacks = "row_deletion", intensity = "fast")
score_survival(result$attacks, result$smallest_kill)
```

`smallest_kill` *Extract the smallest kill from an attack result*

Description

Returns the headline perturbation that killed the claim, or `NULL` when no attack killed the claim in the run.

Usage

```
smallest_kill(result)
```

Arguments

`result`	A `falsifyr_attack` object returned by `attack()`.

Value

A list describing the smallest kill, or `NULL`.

Examples

```
fit <- lm(score ~ treatment + age + baseline_score, data = fragile_trial)
result <- attack(fit, term = "treatment", attacks = "row_deletion", intensity = "fast")
smallest_kill(result)
```

Index