Package 'IncrementalityTEST'

August 7, 2026

Title Analyze Incrementality Experiments

Version 0.1.1

Description Tools for calculating commerce metrics, pairing treatment and control results from A/B testing experiments, estimating incremental effects, and quantifying uncertainty with Student's t and nonparametric bootstrap confidence intervals. Includes validation helpers, a high-level analysis workflow, and compatibility functions for the original package interface.

License Apache License (>= 2)

URL https://github.com/vkobayashi/IncrementalityTEST

BugReports https://github.com/vkobayashi/IncrementalityTEST/issues

Depends R (>= 4.1.0)

Suggests boot (>= 1.3-18), knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Vladimer Kobayashi [aut, cre] (ORCID: <https://orcid.org/0000-0001-9285-9274>)

Maintainer Vladimer Kobayashi <vladimer.kobayashi@gmail.com>

Repository CRAN

Date/Publication 2026-08-07 16:30:25 UTC

Contents

`analyze_incrementality` *Analyze an incrementality experiment collection*

Description

A high-level workflow that pairs groups, calculates experiment-level differences, and summarizes the overall effect with t and bootstrap confidence intervals.

Usage

```
analyze_incrementality(
  data,
  metric,
  id_col = "experiment",
  group_col = "group",
  control = "control",
  treatment = "treatment",
  direction = c("treatment-control", "control-treatment"),
  na_action = c("error", "omit"),
  conf_level = 0.95,
  bootstrap_times = 2000L,
  seed = NULL
)
```

Arguments

`data`	A data frame containing experiment, group, and metric columns.
`metric`	Character scalar naming the metric column.
`id_col`	Character scalar naming the experiment identifier column.
`group_col`	Character scalar naming the group column.
`control`	Value identifying the control group.
`treatment`	Value identifying the treatment group.
`direction`	Difference direction: `"treatment-control"` or `"control-treatment"`.
`na_action`	How to handle missing metric values: `"error"` or `"omit"`.
`conf_level`	Confidence level used by both interval estimators.

bootstrap_times	Number of bootstrap replicates.
seed	Optional bootstrap seed.

Value

An object of class `incrementality_analysis`.

Examples

```
results <- data.frame(
  experiment = rep(LETTERS[1:4], each = 2),
  group = rep(c("control", "treatment"), 4),
  revenue_per_user = c(10, 11, 8, 8.8, 12, 13, 9, 10.5)
)
analyze_incrementality(
  results, "revenue_per_user",
  bootstrap_times = 199, seed = 42
)
```

---

bootstrap_incrementality

Bootstrap confidence interval for a mean effect

Description

Uses ordinary nonparametric resampling of experiment-level differences.

Usage

```
bootstrap_incrementality(
  x,
  times = 2000L,
  conf_level = 0.95,
  type = "percentile",
  seed = NULL,
  na_rm = TRUE
)
```

Arguments

x	Numeric vector of differences.
times	Number of bootstrap replicates.
conf_level	Confidence level between 0 and 1.
type	Interval type. Currently percentile intervals are supported.
seed	Optional integer seed. The caller's random-number state is restored after the function returns.
na_rm	Logical; remove missing values?

Value

An object of class `incrementality_bootstrap`, represented as a list with the estimate, standard error, interval, and bootstrap replicates.

Examples

```
bootstrap_incrementality(c(0.4, 0.8, 0.1, 0.6), times = 199, seed = 1)
```

`calculate_metrics` *Calculate commerce metrics*

Description

Calculates revenue per user (RPU), buyer rate (BR), average order value (AOV), and transactions per buyer (TPB). Inputs are recycled using base R rules, so the function works with individual totals or equally sized vectors.

Usage

```
calculate_metrics(
  nb_transactions,
  nb_users,
  revenue,
  nb_buyers,
  zero_denominator = c("na", "allow")
)
```

Arguments

`nb_transactions`	Numeric vector. Number of transactions.
`nb_users`	Numeric vector. Number of unique users.
`revenue`	Numeric vector. Revenue.
`nb_buyers`	Numeric vector. Number of buyers.
`zero_denominator`	How to handle division by zero: return `NA` (default) or allow R to return infinite/undefined values.

Value

A data frame with columns `RPU`, `BR`, `AOV`, and `TPB`.

Examples

```
calculate_metrics(
  nb_transactions = 11278,
  nb_users = 297073,
  revenue = 279480.4,
  nb_buyers = 10909
)
```

`incrementality_func` *Statistic for the bootstrap (legacy interface)*

Description

Statistic for the bootstrap (legacy interface)

Usage

```
incrementality_func(datadiff, indices)
```

Arguments

`datadiff`	Numeric vector of differences.
`indices`	Resampled indices.

Value

Mean and estimated variance of the mean.

`incrementality_metrics`
Calculate commerce metrics (legacy interface)

Description

This compatibility wrapper returns the same named list as the original package API. New code should generally use `calculate_metrics()`.

Usage

```
incrementality_metrics(nb_transactions, nb_users, revenue, nb_buyers)
```

Arguments

nb_transactions	Numeric vector. Number of transactions.
nb_users	Numeric vector. Number of unique users.
revenue	Numeric vector. Revenue.
nb_buyers	Numeric vector. Number of buyers.

Value

A named list containing RPU, BR, AOV, and TPB.

Examples

```
incrementality_metrics(11278, 297073, 279480.4, 10909)
```

---

legacy_helpers *Legacy vector summary helpers*

Description

These small helpers are retained for compatibility. They return NA when all values are missing. fnFreqDay() returns the first mode when tied.

Usage

```
fnFreqDay(x)

fnMax(x)

fnAve(x)

fnSum(x)
```

Arguments

x	A vector.

Value

fnAve(), fnMax(), and fnSum() return a numeric scalar. fnFreqDay() returns the name of the most frequent value, as a character scalar.

`metric_differences` *Pair experiment groups and calculate metric differences*

Description

For every experiment, pairs one control value with one treatment value and computes `treatment - control` by default. Each experiment must contain exactly one row for each requested group.

Usage

```
metric_differences(
  data,
  metric,
  id_col = "experiment",
  group_col = "group",
  control = "control",
  treatment = "treatment",
  direction = c("treatment-control", "control-treatment"),
  na_action = c("error", "omit")
)
```

Arguments

`data`	A data frame containing experiment, group, and metric columns.
`metric`	Character scalar naming the metric column.
`id_col`	Character scalar naming the experiment identifier column.
`group_col`	Character scalar naming the group column.
`control`	Value identifying the control group.
`treatment`	Value identifying the treatment group.
`direction`	Difference direction: `"treatment-control"` or `"control-treatment"`.
`na_action`	How to handle missing metric values: `"error"` or `"omit"`.

Value

A data frame with the experiment ID, control and treatment values, and a `difference` column.

Examples

```
results <- data.frame(
  experiment = rep(c("A", "B", "C"), each = 2),
  group = rep(c("control", "treatment"), 3),
  RPU = c(10, 11, 8, 9.5, 12, 11.5)
)
metric_differences(results, "RPU")
```

res_boot *Run a bootstrap (legacy interface)*

Description

Requires the suggested boot package and returns a boot object.

Usage

```
res_boot(mydatadiff, statisticfunc = incrementality_func, nb_boot = 2000L)
```

Arguments

mydatadiff	Numeric vector of differences.
statisticfunc	Statistic function accepted by boot::boot().
nb_boot	Number of bootstrap samples.

Value

An object returned by boot::boot().

test_metric *Calculate metric differences (legacy interface)*

Description

Compatibility wrapper for datasets containing iabtest_id and abt_group, where control is coded 0 and treatment is coded 1. It preserves the original convention of control minus treatment.

Usage

```
test_metric(mydata, metric)
```

Arguments

mydata	A data frame containing iabtest_id, abt_group, and the selected metric.
metric	Character scalar naming the metric.

Value

A two-column data frame with iabtest_id and inc_<metric>.

t_confidence_interval *Student's t confidence interval for a mean effect*

Description

Student's t confidence interval for a mean effect

Usage

```
t_confidence_interval(x, conf_level = 0.95, na_rm = TRUE)
```

Arguments

x	Numeric vector of experiment-level differences.
conf_level	Confidence level between 0 and 1.
na_rm	Logical; remove missing values?

Value

A one-row data frame with the sample size, mean, standard error, confidence level, and lower and upper confidence limits.

Examples

```
t_confidence_interval(c(0.4, 0.8, 0.1, 0.6))
```

t_test_cf *Student's t confidence interval (legacy interface)*

Description

Student's t confidence interval (legacy interface)

Usage

```
t_test_cf(datawithID, confinter = 0.05)
```

Arguments

datawithID	A data frame whose second column contains differences.
confinter	Legacy lower-tail alpha value. For example, `0.05` produces a 95 percent two-sided confidence interval.

Value

A list containing the confidence interval, mean, and standard error.

Index