Package: undidR (via r-universe)

November 23, 2026

Title Difference-in-Differences with Unpoolable Data

Version 3.0.3

Maintainer Eric Jamieson <ericbrucejamieson@gmail.com>

Description A framework for estimating difference-in-differences with unpoolable data, based on Karim, Webb, Austin, and Strumpf (2025) <doi:10.48550/arXiv.2403.15910>. Supports common or staggered adoption, multiple groups, and the inclusion of covariates. Also computes p-values for the aggregate average treatment effect on the treated via the randomization inference procedure described in MacKinnon and Webb (2020) <doi:10.1016/j.jeconom.2020.04.024>.

License MIT + file LICENSE

Depends R (>= 4.0)

Imports graphics, grDevices, stats, utils

Encoding UTF-8

RoxygenNote 7.3.2

URL https://github.com/ebjamieson97/undidR, https://ebjamieson97.github.io/undidR/

BugReports https://github.com/ebjamieson97/undidR/issues

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Eric Jamieson [aut, cre, cph]

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

Date/Publication 2026-11-23 03:49:15 UTC

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

RemoteRef HEAD

RemoteSha c5519f5714882c1ddebaad35c8d59038afe31d04

Contents

coef.UnDiDObj 2
create_diff_df 3
create_init_csv 4
plot.UnDiDObj 5
print.UnDiDObj 6
silo71 7
summary.UnDiDObj 7
undid_date_formats 8
undid_stage_three 8
undid_stage_two 10

Index **13**

`coef.UnDiDObj` *Extract coefficients from* `UnDiDObj`

Description

Extract coefficients from `UnDiDObj`

Usage

```
## S3 method for class 'UnDiDObj'
coef(object, level = c("agg", "sub"), ...)
```

Arguments

`object`	A `UnDiDObj` object
`level`	Specify either `"agg"` or `"sub"` to view the aggregate or sub-aggregate results.
`...`	other arguments

Value

A data frame of coefficient estimates

create_diff_df *Creates the* empty_diff_df.csv

Description

Creates the empty_diff_df.csv which lists all of the differences that need to calculated at each silo in order to compute the aggregate ATT. The empty_diff_df.csv is then to be sent out to each silo to be filled out.

Usage

```
create_diff_df(
  init_filepath,
  date_format,
  freq,
  covariates = FALSE,
  freq_multiplier = FALSE,
  weights = "both",
  filename = "empty_diff_df.csv",
  filepath = tempdir()
)
```

Arguments

init_filepath	A character filepath to the init.csv.
date_format	A character specifying the date format used in the init.csv. Call undid_date_formats() to see a list of valid date formats.
freq	A character indicating the length of the time periods to be used when computing the differences in mean outcomes between periods at each silo. Options are: "yearly", "monthly", "weekly", or "daily".
covariates	A character vector specifying covariates to be considered at each silo. If FALSE (default) uses covariates from the init.csv.
freq_multiplier	A numeric value or FALSE (default). Specify if the frequency should be multiplied by a non-zero integer.
weights	A character indicating the weighting to use. The options are "none", "diff", "att", and "both". The options reflect the levels at which weights are applied. "diff" uses weights based off of the number of observations (treated and untreated) when calculating the subaggregate ATTs. "att" uses weights based off of the number of treated observations associated with each subaggregate ATT when calculating the aggregate ATT. "both" applies weighting at both levels, and "none" does not use weights at all. Defaults to "both".
filename	A character filename for the created CSV file. Defaults to "empty_diff_df.csv"
filepath	Filepath to save the CSV file. Defaults to tempdir().

Details

Ensure that dates in the `init.csv` are entered consistently in the same date format. Call `undid_date_formats()` to see a list of valid date formats. Covariates specified when calling `create_diff_df()` will override any covariates specified in the `init.csv`.

Value

A data frame detailing the silo and time combinations for which differences must be calculated in order to compute the aggregate ATT. A CSV copy is saved to the specified directory which is then to be sent out to each silo.

Examples

```
file_path <- system.file("extdata/staggered", "init.csv",
                         package = "undidR")
create_diff_df(
  init_filepath = file_path,
  date_format = "yyyy",
  freq = "yearly"
)
unlink(file.path(tempdir(), "empty_diff_df.csv"))
```

---

create_init_csv *Creates the* `init.csv`

Description

The `create_init_csv()` function generates a CSV file with information on each silo's start times, end times, and treatment times. If parameters are left empty, generates a blank CSV with only the headers.

Usage

```
create_init_csv(
  silo_names = character(),
  start_times = character(),
  end_times = character(),
  treatment_times = character(),
  covariates = character(),
  filename = "init.csv",
  filepath = tempdir()
)
```

Arguments

`silo_names`	A character vector of silo names.
`start_times`	A character vector of start times.
`end_times`	A character vector of end times.
`treatment_times`	A character vector of treatment times.
`covariates`	A character vector of covariates, or, `FALSE` (default).
`filename`	A character filename for the created initializing CSV file. Defaults to `"init.csv"`.
`filepath`	Filepath to save the CSV file. Defaults to `tempdir()`.

Details

Ensure dates are entered consistently in the same date format. Call `undid_date_formats()` to view valid date formats. Control silos should be marked as `"control"` in the `treatment_times` vector. If `covariates` is `FALSE`, no covariate column will be included in the CSV.

Value

A data frame containing the contents written to the CSV file. The CSV file is saved in the specified directory (or in a temporary directory by default) with the default filename `init.csv`.

Examples

```
create_init_csv(
  silo_names = c("73", "46", "54", "23", "86", "32",
                 "71", "58", "64", "59", "85", "57"),
  start_times = "1989",
  end_times = "2000",
  treatment_times = c(rep("control", 6),
                      "1991", "1993", "1996", "1997", "1997", "1998"),
  covariates = c("asian", "black", "male")
)
unlink(file.path(tempdir(), "init.csv"))
```

`plot.UnDiDObj` *Plot method for* `UnDiDObj`

Description

Plot method for `UnDiDObj`

Usage

```
## S3 method for class 'UnDiDObj'
plot(
  x,
  event = FALSE,
  event_window = NULL,
  ci = 0.95,
  lwd = 1,
  legend = "topright",
  ...
)
```

Arguments

x	A UnDiDObj object
event	Logical. If TRUE, creates an event study plot. If FALSE (default), creates a parallel trends plot.
event_window	Numeric vector of length 2 specifying the event window as c(start, end). Default is NULL (use all available periods).
ci	Numeric between 0 and 1 specifying confidence level. Default is 0.95.
lwd	Linewidth arg passed to lines(), abline(), and segments(). Defaults to 1.
legend	Keywords for indicating desired legend location. Defaults to "topright". Other options include any of the keywords used as x in legend(x, ...) or NULL to omit a legend.
...	other arguments passed to plot

---

print.UnDiDObj *Print method for* UnDiDObj

Description

Print method for UnDiDObj

Usage

```
## S3 method for class 'UnDiDObj'
print(x, level = c("agg", "sub"), ...)
```

Arguments

x	A UnDiDObj object.
level	Specify either "agg" or "sub" to view the aggregate or sub-aggregate results.
...	other arguments

silo71 *Example merit data*

Description

A dataset containing college enrollment and demographic data for analyzing the effects of merit programs in state 71.

Usage

```
silo71
```

Format

A tibble with 569 rows and 7 variables:

coll Binary indicator for college enrollment (outcome variable)

merit Binary indicator for merit program (treatment variable)

male Binary indicator for male students

black Binary indicator for Black students

asian Binary indicator for Asian students

year Year of observation

state State identifier

summary.UnDiDObj *Summary method for* UnDiDObj

Description

Summary method for UnDiDObj

Usage

```
## S3 method for class 'UnDiDObj'
summary(object, level = c("all", "agg", "sub"), ...)
```

Arguments

object	A UnDiDObj object
level	Specify either "agg", "sub", or "all", to view the results at the aggregate level, the sub-aggregate level, or to view both simultaneously.
...	other arguments

undid_date_formats *Shows valid date formats*

Description

The undid_date_formats() function returns a list of all valid date formats that can be used within the undidR package.

Usage

```
undid_date_formats()
```

Details

The date formats returned by this function are used to ensure consistency in date processing within the undidR package.

Value

A named list containing valid date formats:

- General_Formats: General date formats compatible with the package.
- R_Specific_Formats: Date formats specific to R.
- Other_Formats: Formats seen sometimes in Stata.

Examples

```
undid_date_formats()
```

undid_stage_three *Computes UNDID results*

Description

Takes in all of the filled diff df CSV files and uses them to compute group level ATTs as well as the aggregate ATT and its standard errors and p-values. Also takes in the trends data CSV files and uses them to produce parallel trends and event study plots.

Usage

```
undid_stage_three(
  dir_path,
  agg = "g",
  weights = "both",
  covariates = FALSE,
  notyet = FALSE,
  nperm = 999,
  verbose = 100,
  check_anon_size = FALSE,
  hc = "hc1",
  only = NULL,
  omit = NULL,
  max_attempts = 100
)
```

Arguments

dir_path	A character specifying the filepath to the folder containing all of the filled diff df CSV files.
agg	A character which specifies the aggregation methodology for computing the aggregate ATT in the case of staggered adoption. Options are: "silo", "g", "gt", "sgt", "time", "none". Defaults to "g". "silo" computes a subaggregate ATT for each silo, "g" computes a subaggregate ATT for each unique treatment time, "gt" computes a subaggregate ATT for each unique treatment time & post-treatment period pair, "sgt" computes a subaggregate ATT for each treatment time & post-treatment pair, separated by silo, "time" computes subaggregate ATTs for grouped by time since treatment, and "none" does not compute any subaggregate ATTs, but rather computes an aggregate ATT directly from the differences.
weights	A string, determines which of the weighting methodologies should be used. Options are: "none", "diff", "att", or "both". Defaults to the weighting choice specified in the filled diff CSV files.
covariates	A logical value (either TRUE or FALSE) which specifies whether to use the diff_estimate column or the diff_estimate_covariates column from the filled diff df CSV files when computing ATTs.
notyet	A logical value which declares if the not-yet-treated differences from treated silos should be used as controls when computing relevant sub-aggregate ATTs. Defaults to FALSE.
nperm	Number of random permutations of treatment assignment to use when calculating the randomization inference p-value. Defaults to 999.
verbose	A numeric value (or NULL) which toggles messages showing the progress of the randomization inference once every verbose iterations. Defaults to 100.
check_anon_size	A logical value, which if TRUE displays which silos enabled the anonymize_weights argument in stage two, and their respective anonymize_size values. Defaults to FALSE.

hc	Specify which heteroskedasticity-consistent covariance matrix estimator (HC-CME) should be used. Options are 0, 1, 2, 3, and 4 (or "hc0", "hc1", "hc2", "hc3", "hc4"). Defaults to "hc1".
only	A character vector of silos to include. Defaults to NULL.
omit	A character vector of silos to omit. Defaults to NULL.
max_attempts	A numeric value. Sets the maximum number of attempts to find a new unique random permutations during the randomization inference procedure. Defaults to 100.

Value

An UnDiDObj with S3 methods of summary(), plot(), print(), and coef().

Examples

```
# Execute `undid_stage_three()`
dir <- system.file("extdata/staggered", package = "undidR")

   # Recommended: nperm >= 399 for reasonable precision
   # (~15 seconds on typical hardware)
   result <- undid_stage_three(dir, agg = "g", nperm = 399, verbose = NULL)

   # View the summary of results
   summary(result)

   # View the parallel trends plot
   plot(result)

   # View the event study plot
   plot(result, event = TRUE)
```

---

undid_stage_two *Runs UNDID stage two procedures*

Description

Based on the information given in the received empty_diff_df.csv, computes the appropriate differences in mean outcomes at the local silo and saves as filled_diff_df_$silo_name.csv. Also stores trends data as trends_data_$silo_name.csv.

Usage

```
undid_stage_two(
  empty_diff_filepath,
  silo_name,
  silo_df,
  time_column,
```

```
  outcome_column,
  silo_date_format = NULL,
  consider_covariates = TRUE,
  filepath = tempdir(),
  anonymize_weights = FALSE,
  anonymize_size = 5,
  factor = NULL
)
```

Arguments

empty_diff_filepath	A character filepath to the empty_diff_df.csv.
silo_name	A character indicating the name of the local silo. Ensure spelling is the same as it is written in the empty_diff_df.csv.
silo_df	A data frame of the local silo's data. Ensure any covariates are spelled the same in this data frame as they are in the empty_diff_df.csv.
time_column	A character which indicates the name of the column in the silo_df which contains the date data. Ensure the time_column references a column of character values.
outcome_column	A character which indicates the name of the column in the silo_df which contains the outcome of interest. Ensure the outcome_column references a column of numeric values.
silo_date_format	A character which indicates the date format which the date strings in the time_column are written in.
consider_covariates	An optional logical parameter which if set to FALSE ignores any of the computations involving the covariates. Defaults to TRUE.
filepath	Character value indicating the filepath to save the CSV files. Defaults to tempdir().
anonymize_weights	A logical value (defaults FALSE) which determines if the counts of n (# of obs. used to calculate a contrast/difference) and n_t (# of treated obs. used in the calculation of a contrast/difference) should be rounded.
anonymize_size	A numeric value. Counts will be rounded to the nearest multiple of this value if anonymize_weights is TRUE (with a minimum value for any count being set as the value given for anonymize_size).
factor	An optional character vector (e.g. c("educ", "race")) or formula(s) (e.g. ~educ + race, ~educ ~race, or c(~educ, ~race)) naming covariates that should be treated as categorical even though they are not character or factor columns, such as numerically coded categories. Character and factor covariates are always treated as categorical. Defaults to NULL.

Details

Covariates at the local silo should be renamed to match the spelling used in the empty_diff_df.csv.

Covariates may be numeric, logical, character, or factor columns. Character and factor covariates (and any covariates named in `factor`) are treated as categorical and are automatically converted into dummy variables, omitting one reference category. For factor columns, unused levels are dropped and the reference category is the lowest remaining level. For all other categorical columns, the categories are ordered with `sort(method = "radix")` and the first is used as the reference.

Rows with missing values in the time, outcome, or covariate columns are dropped with a warning that reports the number of missing values in each affected column. Values in character or factor columns that are empty or contain only whitespace are treated as missing. Other values are not modified (e.g. leading or trailing whitespace is not removed).

Value

A list of data frames. The first being the filled differences data frame, and the second being the trends data data frame. Use the suffix $diff_df to access the filled differences data frame, and use $trends_data to access the trends data data frame.

Examples

```
# Load data
silo_data <- silo71
empty_diff_path <- system.file("extdata/staggered", "empty_diff_df.csv",
                               package = "undidR")

# Run `undid_stage_two()`
results <- undid_stage_two(
  empty_diff_filepath = empty_diff_path,
  silo_name = "71",
  silo_df = silo_data,
  time_column = "year",
  outcome_column = "coll",
  silo_date_format = "yyyy"
)

# View results
head(results$diff_df)
head(results$trends_data)

# Clean up temporary files
unlink(file.path(tempdir(), c("diff_df_71.csv",
                              "trends_data_71.csv")))
```

Index

∗ **datasets**
silo71, 7

coef.UnDiDObj, 2
create_diff_df, 3
create_init_csv, 4

plot.UnDiDObj, 5
print.UnDiDObj, 6

silo71, 7
summary.UnDiDObj, 7

undid_date_formats, 8
undid_date_formats(), *3–5*
undid_stage_three, 8
undid_stage_two, 10