Package 'undidR'

January 23, 2025

Title Difference-in-Differences with Unpoolable Data

Version 1.0.0

Maintainer Eric Jamieson <ericbrucejamieson@gmail.com>

Description A framework for estimating difference-in-differences with unpoolable data, based on Karim, Webb, Austin, and Strumpf (2024) <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 CRAN

Date/Publication 2025-01-23 17:30:05 UTC

Contents

`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 = "standard",
  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 in the case of common adoption. The `"standard"` (default) weight is calculated as $w_s = \frac{N_s^{\text{post}}}{N_s^{\text{post}}+N_s^{\text{pre}}}$. Options are: `"standard"`.
`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_parallel_trends` *Plots parallel trends figures*

Description

The `plot_parallel_trends()` function combines the various trends data CSV files and plots parallel trends figures. All treatment and all control groups can be combined so that there is one control line and one treatment line by setting `combine = TRUE`.

Usage

```
plot_parallel_trends(
  dir_path,
  covariates = FALSE,
  save_csv = FALSE,
  combine = FALSE,
  pch = NA,
  pch_control = NA,
  pch_treated = NA,
  control_colour = c("darkgrey", "lightgrey"),
  control_color = NULL,
  treatment_colour = c("darkred", "lightcoral"),
  treatment_color = NULL,
  lwd = 2,
  xlab = NA,
  ylab = NA,
  title = NA,
  xticks = 4,
  date_format = "%Y-%m-%d",
  xdates = NULL,
  xaxlabsz = 0.8,
  save_png = FALSE,
  width = 800,
  height = 600,
  ylim = NULL,
  yaxlabsz = 0.8,
  ylabels = NULL,
  yticks = 4,
  ydecimal = 2,
  legend_location = "topright",
  simplify_legend = TRUE,
  legend_cex = 0.7,
  legend_on = TRUE,
  treatment_indicator_col = "grey",
  treatment_indicator_alpha = 0.5,
  treatment_indicator_lwd = 2,
  treatment_indicator_lty = 2,
  interpolate = FALSE,
  filepath = tempdir(),
  filenamecsv = "combined_trends_data.csv",
  filenamepng = "undid_plot.png"
)
```

Arguments

`dir_path`	A character filepath to the folder containing all of the trends data CSV files.
`covariates`	A logical value (defaults to `FALSE`) indicating whether or not to consider covariates, i.e. whether or not to use the `mean_outcome` column or the `mean_outcome_residualized`

column from the trends data CSV files.

save_csv A logical value (defaults to FALSE) indicating whether or not to save the combined_trends_data.csv.

combine A logical value (defaults to FALSE) indicating whether to plot each silo separately or to combine silos based on treatment status.

pch An integer (0 to 25) or vector of integers (from 0 to 25) which determine the style of points used on the plot. Setting to NA (default) will omit points from the plot.

pch_control An integer (from 0 to 25) or vector of integers (from 0 to 25) which determine the style of points used on the plot for control silos. Takes value of pch if set to NULL (default).

pch_treated An integer (from 0 to 25) or vector of integers (from 0 to 25) which determine the style of points used on the plot for treated silos. Takes value of pch if set to NULL (default).

control_colour A character vector of colours (defaults to c("darkgrey", "lightgrey")) for the control silo lines. If combine = TRUE, takes the 1st value to determine the colour of the control line.

control_color Overrides control_colour if used. Defaults to NULL.

treatment_colour
A character vector of colours (defaults to c("darkred", "lightcoral")) for the treatment silos. If combine = TRUE, takes the 1st value to determine the colour of the control line.

treatment_color
Overrides control_colour if used. Defaults to NULL.

lwd An integer (defaults to 2) for selecting the line widths.

xlab A character value for the x-axis label (defaults to NA).

ylab A character value for the y-axis label (defaults to NA).

title A character value for the title of the plot (defaults to NA).

xticks An integer value denoting how many ticks to display on the x-axis (defaults to 4).

date_format A string value denoting the format with which to display the dates along the x-axis (defaults to "%Y"). Uses standard R date formatting styles.

xdates Takes in a vector of date objects to be used as the dates shown along the x-axis (defaults to NULL).

xaxlabsz A double indicating the x-axis label sizes in comparison to a standardized default size (defaults to 0.8).

save_png A logical value indicating whether or not to save the plot as a PNG file (defaults to FALSE).

width An integer denoting the width of the saved PNG file.

height An integer denoting the height of the saved PNG file.

ylim A vector of two doubles defining the min and max range of the values on the y-axis. Defaults to the min and max values of the values to be plotted.

yaxlabsz A double for specifying the y-axis label sizes (defaults to 0.8) in comparison to a standardized default size.

`ylabels` A vector of values that you would like to appear on the y-axis (defaults to `NULL`).

`yticks` An integer denoting how many values to display along the y-axis (defaults to `4`).

`ydecimal` An integer value denoting to which decimal point the values along the y-axis are rounded to.

`legend_location`
A character value for determining the location of the legend (defaults to `"topright"`). Options are: `"topright"`, `"topleft"`, `"bottomright"`, `"bottomleft"`, `"top"`, `"bottom"`, `"left"`, `"right"`, `"center"`.

`simplify_legend`
A logical value which if set to `TRUE` shows one colour for the treatment silos in the legend and one colour for the control silos. Defaults to `TRUE`.

`legend_cex` A double for adjusting the size of the text in the legend compared to a standard default size. Defaults to `0.7`.

`legend_on` A logical value for turning the legend on or off (defaults to `TRUE`).

`treatment_indicator_col`
A character value for determining the colour of the dashed vertical lines showing when treatment times were (defaults to `"grey"`).

`treatment_indicator_alpha`
A double for for determining the transparency level of the dashed vertical lines showing the treatment times (defaults to `0.5`).

`treatment_indicator_lwd`
A double for selecting the line width of the treatment indicator lines (defaults to `2`).

`treatment_indicator_lty`
An integer for the selecting the lty option, i.e. the line style, for the treatment_indicator lines (defaults to `2`).

`interpolate` A logical value (either `TRUE` or `FALSE`) which determines if interpolation should be used to fill missing trends data. Defaults to `FALSE`. Uses a piecewise linear function.

`filepath` Filepath to save the CSV file. Defaults to `tempdir()`.

`filenamecsv` A string filename for the combined trends data Defaults to `"combined_trends_data.csv"`.

`filenamepng` A string filename for the PNG file output. Defaults to `"undid_plot.png"`.

Value

A data frame built from the trends data from all CSV files in the specified directory. If `combine = FALSE`, the data frame includes all silos joined by row. If `combine = TRUE`, the data frame merges treated silos into a single treatment group and control silos into a single control group.

Examples

```
# Get path to example data included with package
dir_path <- system.file("extdata/staggered", package = "undidR")

# Basic usage with default parameters
plot_parallel_trends(dir_path)
```

```
# Custom plot with modified parameters
plot_parallel_trends(dir_path, combine = TRUE, lwd = 4,
                     xdates = as.Date(c("1989-01-01", "1991-01-01",
                                        "1993-01-01", "1995-01-01",
                                        "1997-01-01", "1999-01-01")))
```

`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

Source

https://economics.uwo.ca/people/conley_docs/code_to_download.html

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.

Usage

```
undid_stage_three(
  dir_path,
  agg = "silo",
  weights = TRUE,
  covariates = FALSE,
  interpolation = FALSE,
  save_csv = FALSE,
  filename = "UNDID_results.csv",
  filepath = tempdir(),
  nperm = 1001,
  verbose = TRUE
)
```

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", or "gt". Defaults to "silo".
weights	A logical value (either TRUE or FALSE) which determines whether or not the weights should be used in the case of common adoption. Defaults to TRUE.
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.
interpolation	A logical value or a character which specifies which, if any, method of interpolation/extrapolation for missing values of diff_estimate or diff_estimate_covariates should be used. There must be at least one diff_estimate or diff_estimate_covariates value for the (silo,g) group for which a missing value is being estimated in order for interpolation to work. Options are: "linear_function", "nearest_value", or "piecewise_linear". Defaults to FALSE.
save_csv	A logical value, either TRUE or FALSE (default), which determines if a CSV copy of the UNDID results will be saved or not.
filename	A string filename for the created CSV file. Defaults to "UNDID_results.csv"
filepath	Filepath to save the CSV file. Defaults to tempdir().
nperm	Number of random permutations of gvar & silo pairs to consider when calculating the randomization inference p-value. Defaults to 1001.
verbose	A logical value (either TRUE or FALSE) which toggles messages showing the progress of the randomization inference. Defaults to TRUE.

Details

The agg parameter specifies the aggregation method used in the case of staggered adoption. By default it is set to "silo" so that the ATTs are aggregated across silos with each silo having equal weight, but can be set to "gt" or "g" instead. Aggregating across "g" calculates ATTs for groups

based on when the treatment time was, with each "g" group having equal weight. Aggregating across "gt" calculates ATTs for groups based on when the treatment time was and the time for which the ATT is calculated. The agg parameter is ignored in the case of a common treatment time and only takes effect in the case of staggered adoption. For common adoption, refer to the weights parameter.

Value

A data frame containing the aggregate ATT and its standard errors and p-values from two-sided tests of agg_ATT == 0. Also returns group (silo, g, or gt) level ATTs for staggered adoption.

Examples

```
# Execute `undid_stage_three()`
dir <- system.file("extdata/staggered", package = "undidR")
undid_stage_three(dir, agg = "g", nperm = 501, verbose = FALSE)
```

---

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,
  consider_covariates = TRUE,
  filepath = tempdir()
)
```

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()`.

Details

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

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