DiDSL - A Domain Specific Language for Diff-in-Diff Analysis

1 Introduction

DiDSL is a domain specific language for Difference in Difference (DiD) analysis. It provides a set of commands enabling data preparation, transformation, assumption verification, and model estimation steps in DiD analysis. Take a look at the following DiDSL example script:

```
load dataset "standard_did_simulated.csv" as standard_did
set id column "id"
set group column "treat" with treatment values 1
set time column "time" pre-period -5 -4 -3 -2 -1 post-period 0
set outcome column "y"
check parallel trends on standard_did
run did regression on standard_did
run did fixed effects on standard_did
```

This is what the commands in the script above do:

- `load dataset "standard_did_simulated.csv" as standard_did`
  Loads the CSV file `standard_did_simulated.csv` into memory and names it `standard_did`
- `set id column "id"`
  Defines the `id` column as unit identifier.
- `set group column "treat" with treatment values 1`
  Defines the `treat` column as the group identifier, marking 1 as the treated group.
- `set time column "time" pre-period -5 -4 -3 -2 -1 post-period 0`
  Defines the `time` column as the time variable, specifying which values correspond to pre-treatment and post-treatment periods.
- `set outcome column "y"`
  Sets the outcome variable to `y` (e.g., total employment).
- `check parallel trends on`
  Runs a test to verify if treatment and control groups had similar trends before treatment.

- `run did regression`
  Estimates the difference-in-differences regression model on the `y` data.

- `run did fixed effects using robust`
  Runs a Difference-in-Differences regression on the specified dataset using fixed effects for the individual units (defined by the `id` column) and time periods. Including robust standard errors helps to obtain consistent inference in the presence of heteroskedasticity or within-cluster correlation.

The output of the above script is the following one:

```
Loading data from 'standard_did_simulated.csv' as 'standard_did'...
Data 'standard_did' loaded successfully. standard_did
ID column for 'standard_did' set to 'id'.
Treatment column for 'standard_did' set to 'treat' with value '1'.
Time column for 'standard_did' set to 'time'. Pre-period: -5--1 (from [-5, -4, -3, -2, -1]),
Post-period: 0-0 (from [0]).
Outcome column for 'standard_did' set to 'y'.

 Parallel Trends Test | p-threshold = 0.05
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.001
Model:                            OLS   Adj. R-squared:                 -0.001
Method:                 Least Squares   F-statistic:                    0.3541
Date:                Mon, 28 Jul 2025   Prob (F-statistic):              0.786
Time:                        23:05:27   Log-Likelihood:                -2646.0
No. Observations:                1500   AIC:                             5300.
Df Residuals:                    1496   BIC:                             5321.
Df Model:                           3
Covariance Type:            nonrobust
================================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Intercept                    0.1545      0.089      1.727      0.084      -0.021       0.330
treatment                   -0.1161      0.126     -0.918      0.359      -0.364       0.132
time_numeric                -0.0263      0.037     -0.721      0.471      -0.098       0.045
treatment:time_numeric       0.0527      0.052      1.020      0.308      -0.049       0.154
==============================================================================
Omnibus:                        0.449   Durbin-Watson:                   1.978
Prob(Omnibus):                  0.799   Jarque-Bera (JB):                0.406
Skew:                           0.039   Prob(JB):                        0.816
Kurtosis:                       3.021   Cond. No.                         12.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
 Parallel trends assumption holds (p = 0.3078)
Running Difference-in-Differences analysis for 'standard_did'...
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.480
Model:                            OLS   Adj. R-squared:                  0.479
Method:                 Least Squares   F-statistic:                     552.9
Date:                Mon, 28 Jul 2025   Prob (F-statistic):          1.78e-254
Time:                        23:05:27   Log-Likelihood:                -3164.1
```

```
No. Observations:                1800   AIC:                             6336.
Df Residuals:                    1796   BIC:                             6358.
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept      0.1018      0.051      1.985      0.047       0.001       0.202
treatment     -0.0107      0.073     -0.148      0.883      -0.153       0.132
post          -0.1701      0.126     -1.354      0.176      -0.417       0.076
interaction    5.0373      0.178     28.345      0.000       4.689       5.386
================================================================================
Omnibus:                        0.140   Durbin-Watson:                   1.979
Prob(Omnibus):                  0.932   Jarque-Bera (JB):                0.138
Skew:                           0.021   Prob(JB):                        0.934
Kurtosis:                       2.996   Cond. No.                         7.25
================================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

--- DiD Results for 'standard_did' ---
DiD effect: 5.0373 (p = 0.0000)
Covariates: on, Robust SEs: off, Clustered SEs: off
PanelOLS formula: y ~ 1 + interaction + EntityEffects + TimeEffects
                          PanelOLS Estimation Summary
================================================================================
Dep. Variable:                          y   R-squared:                        0.5163
Estimator:                   PanelOLS   R-squared (Between):              0.1326
No. Observations:                1800   R-squared (Within):               0.6590
Date:                Mon, Jul 28 2025   R-squared (Overall):              0.4791
Time:                        23:05:27   Log-likelihood                   -2381.2
Cov. Estimator:            Unadjusted
                                        F-statistic:                      1595.0
Entities:                         300   P-value                           0.0000
Avg Obs:                       6.0000   Distribution:                  F(1,1494)
Min Obs:                       6.0000
Max Obs:                       6.0000   F-statistic (robust):             1595.0
                                        P-value                           0.0000
Time periods:                     6     Distribution:                  F(1,1494)
Avg Obs:                       300.00
Min Obs:                       300.00
Max Obs:                       300.00

                             Parameter Estimates
================================================================================
             Parameter  Std. Err.     T-stat    P-value    Lower CI    Upper CI
--------------------------------------------------------------------------------
Intercept       0.0681     0.0257     2.6463     0.0082      0.0176      0.1186
interaction     5.0373     0.1261     39.937     0.0000      4.7898      5.2847
================================================================================

F-test for Poolability: 6.8261
P-value: 0.0000
Distribution: F(304,1494)

Included effects: Entity, Time
```

2 DiDSL Commands

2.1 Data Preparation

2.1.1 Command: `load dataset`

Syntax:

```
load dataset "filename.csv" as dataset_name [with header | without header]
```

Description:
Loads a CSV dataset from the specified file and assigns it an internal name for later use. An optional clause allows explicit control over whether the dataset includes a header row.

Arguments:

- `"filename.csv"` — The name or path of the CSV file to load. Must be enclosed in double quotes.
- `as dataset_name` — The alias to assign to the dataset.
- `with header` (optional) — Indicates that the first row of the file contains column names (default behavior).
- `without header` (optional) — Indicates that the file does not contain column names; default column names like `V1`, `V2`, ... will be assigned.

Examples:

```
load dataset "data.csv" as my_data
load dataset "raw_data.csv" as raw without header
load dataset "cleaned.csv" as cleaned with header
```

Notes:

- If neither `with header` nor `without header` is specified, the system assumes `with header`.
- The dataset must be in comma-separated format (.csv).
- Filenames are interpreted relative to the current working directory.

Command: `show dataset`

Purpose Displays the contents of a dataset currently loaded in memory. Optionally shows the full dataset or just the first few rows.

Syntax:

```
show dataset dataset_name [all]
```

Arguments:

- `dataset_name` — Name of the dataset, as previously loaded using `load dataset`.
- `all` (optional) — If specified, displays the entire dataset. If omitted, only the first 10 rows are shown.

Examples:

```
show dataset my_data
show dataset employment all
```

Behavior:

- By default, prints the first 10 rows.
- If `all` is specified, prints the entire DataFrame using full row formatting.
- If the dataset does not exist, an error message is printed.
- When verbose mode is enabled, diagnostic messages are printed before attempting to show the dataset.

Notes:

- This command is useful for debugging or inspecting the result of transformations.
- The data shown is whatever is currently stored in memory — including after any imputations or updates.

2.1.2 Commands: `set ... column`

Purpose These commands assign structural roles to specific columns in a dataset, such as the treatment group identifier, time variable, outcome, or optional subgroups. This setup is required before most estimation or imputation commands can run.

`set group column`

Syntax:

```
set group column "column_name" on dataset_name with

treatment values "treated_group[, ...]"
```

Description: Specifies which column identifies treatment and control groups in the dataset.

Arguments:

- `"column_name"` – The column containing group identifiers.

- `on dataset_name` – The dataset to configure.
- `with treatment values "..."` – One or more values (as a comma-separated string) that define the treated group(s).

Example:

```
set group column "state" on employment with treatment values "New Jersey"
```

Note: All other values are treated as part of the control group.

`set time column`

Syntax:

```
set time column "column_name" on dataset_name pre-period "val1[, val2, ...]"
post-period "val3[, val4, ...]"
```

Description: Specifies which column defines the time variable, and which values are considered pre-treatment vs. post-treatment periods.

Example:

```
set time column "time" on employment pre-period "Feb-92" post-period "Nov-92"
```

Note: Comma-separated lists (e.g., `"2018,2019"`) are accepted for multi-period cases.

`set outcome column`

Syntax:

```
set outcome column "column_name" on dataset_name
```

Description: Identifies the outcome variable to be analyzed (e.g., total employment, revenue, test scores, etc.).

Example:

```
set outcome column "emptot" on employment
```

Note: Only one outcome column can be set per dataset at a time.

`set subgroup column`

Syntax:

```
set subgroup column "column_name" on dataset_name
```

Description: (Optional) Assigns a column that defines finer sub-groupings within each treatment/control group, such as store chains, regions, or income bands.

Example:

```
set subgroup column "chain" on employment
```

Effect: If used, certain commands (like `impute missing data`) will use this column to compute more granular group-wise statistics.

Optional: If no subgroup column is defined, commands that support it will fall back to using only `group_col` and `time_col`.

```
set covariates
```

Syntax

```
set covariates "col1, col2, ..." on dataset_name
```

Purpose Defines a list of covariate columns to be used in adjusted estimation methods (e.g. regression DiD, doubly robust ATT, etc.).

Arguments

- `"col1, col2, ..."` — A comma-separated list of column names representing observed covariates.
- `dataset_name` — The dataset on which the covariates apply.

Behavior

- These covariates will be used in any subsequent estimation commands that support adjustment — e.g., `run did regression`, `run att estimations`, etc.
- If no covariates are set, these commands will run without adjustment.

Example

```
set covariates "lpop, age, income" on mpdta_prova
```

Notes

- All columns listed must exist in the dataset.
- Whitespace around column names is ignored.
- The covariates are stored and reused automatically by methods that support them — you do not need to specify them again.

2.1.3 Commands: `remove na`

Purpose Removes incomplete pre–post pairs for each unit so that every retained unit has exactly one observed value in each specified pre- and post-period. This ensures the Difference-in-Differences routines always operate on balanced, non-missing comparisons.

Syntax:

```
remove na on dataset_name
    pre-period "pre_val1" "pre_val2" ... "pre_valK"
    post-period "post_val1" "post_val2" ... "post_valK"
```

Arguments

- `dataset_name` — Alias of the dataset (as loaded via `load dataset`).
- `pre-period "..."` — A list of K time values defining the pre-treatment periods, in order.
- `post-period "..."` — A list of K time values defining the corresponding post-treatment periods. Must have the *same length* K as the pre-period list.

Behavior

1. For each unit (identified by the `id` column) and each index $i = 1, \ldots, K$:
   - Locate the row at time `pre_vali` and the row at time `post_vali`.
   - If *either* row is missing or contains `NA` in any of the *key columns* (`group`, `time`, `outcome`, or any `covariates`), *drop both* rows.
   - Otherwise, *keep both* rows.
2. No other rows are affected — only the explicitly paired times are considered.
3. The resulting dataset is guaranteed to have, for each unit and each i, one non-missing pre/post pair.

Requirements

- You must first set
  - the `id` column via `set id column`,
  - the `group` column via `set group column`,
  - the `time` column (even though the `pre-period`/`post-period` values here override it for pairing),
  - the `outcome` column via `set outcome column`,
  - (optionally) any `covariates` via `set covariates`.
- The `pre-period` and `post-period` lists must be of equal length.

Example

```
load dataset "na_simulated.csv" as na_data
set id column "id" on na_data
set group column "treat" on na_data with treatment values 1
set time column "period" on na_data
    pre-period "pre1" "pre2"
    post-period "post1" "post2"
set outcome column "y" on na_data

remove na on na_data
    pre-period "pre1" "pre2"
    post-period "post1" "post2"

show dataset na_data all
```

In this example, any unit for which either `pre1` or `post1` is missing will lose both of those rows; likewise for the $\{pre2, post2\}$ pair. Only fully observed pre/post pairs remain.

2.1.4 Commands: `remove na all`

Purpose Drops *all* observations for any unit (as identified by the `id_column`) that has a missing value in any of the key analysis columns (group, time, outcome, or covariates). This is useful when you require every panel unit to have a fully observed economy of pre- and post-treatment data.

Syntax:

```
remove na all on dataset_name
```

Arguments

- `dataset_name` — The alias of the dataset (as loaded via `load dataset`) on which to operate.

Required prior configuration Before calling this command, you must have already:

- Loaded the dataset via `load dataset`.
- Set the unit identifier column via `set id column "column_name" on dataset_name`.
- Defined the group column via `set group column`.
- Defined the time column (with pre- and post-periods) via `set time column`.
- Defined the outcome column via `set outcome column`.
- (Optional) Defined covariates via `set covariates`.

What it does

1. Gathers the list of columns of interest: `group_col`, `time_col`, `outcome_col`, and any `covariates`.
2. Identifies every unit (by `id_col`) that has at least one `NA` in any of those columns.
3. Drops *all* rows belonging to those units from the in-memory dataset.
4. Overwrites the dataset so that subsequent commands (e.g. regressions, plots) see only fully observed units.

Example

```
load dataset "mpdta_prova.csv" as mpdta_prova
set id column  "countyreal" on mpdta_prova

set group column "first.treat" on mpdta_prova
with treatment values "2004,2006,2007"

set time column  "year" on mpdta_prova pre-period "2003"
post-period "2004,2006,2007"

set outcome column "lemp" on mpdta_prova

set covariates  "lpop" on mpdta_prova

remove na all on mpdta_prova

show dataset mpdta_prova all
```

Notes

- This command is *stricter* than `remove na`: it eliminates entire panels rather than balancing pre vs. post within each unit.
- If `id_column` has not been set, the command will error out.
- Use `remove na all` when you need every treated/control unit to have a complete set of pre- and post-treatment observations for your DiD estimators.

2.1.5 Commands: `impute missing data`

Purpose Replace *NaN / missing* values in a chosen column by the mean of that column computed over an appropriate reference group. The reference group is defined by:

- the *treatment group identifier* (`group_col`) and
- a binary indicator for `pre`- vs. `post`-treatment periods, derived from the `time_col`.
- **Optionally:** an additional *sub-group* column (`subgroup_col`).

Grammar

```
impute missing data for "column_name" on dataset_name
```

Arguments

`"column_name"` (string, required) Name of the column whose missing values will be imputed.

`dataset_name` (identifier, required) Name of the dataset previously loaded with `load dataset`.

Required prior configuration Before calling this command, the following must already be set for `dataset_name`:

1. `group_col` – via `set group column`
2. `time_col` – via `set time column`
3. `pre_period_values` & `post_period_values`
4. **Optionally** `subgroup_col` – via `set subgroup column`

Operational logic

1. Internally create a temporary flag `pre_or_post` = 0 (pre) / 1 (post) based on `time_col` $\in$ {`pre_period_values`, `post_period_values`}.
2. Compute the mean of `"column_name"` in each reference group:

$$\text{Reference group} = \begin{cases} (\texttt{group\_col}, \texttt{pre\_or\_post}) & \text{if no sub-group column} \\ (\texttt{group\_col}, \texttt{pre\_or\_post}, \texttt{subgroup\_col}) & \text{if sub-group column present} \end{cases}$$

3. Merge these means back and plug them in wherever the target cell is `NaN`.
4. Drop all helper columns and overwrite the in-memory dataframe.

Effect of the *sub-group* column If a sub-group column has been declared, the imputation becomes more granular: each missing value is filled with the mean calculated *within its own* `subgroup_col` level. If no sub-group is configured, the command falls back to the coarser `(group_col, pre_or_post)` means; a console message reflects the chosen strategy ("Using subgroup ..." vs. "Using group and time means").

Example

```
set group column "state"   on employment with treatment values "New Jersey"
set time  column "time"    on employment pre-period "Feb-92" post-period "Nov-92"
set subgroup column "chain" on employment

impute missing data for "emptot" on employment
```

The call above will:

- flag each row as pre- or post-treatment,
- compute *mean employment* (`emptot`) for every combination of (`state`, pre/post, `chain`),
- fill in missing `emptot` values with the corresponding mean,
- leave observed (non-missing) values untouched.

Console output When `verbose` is enabled (default), the command prints which strategy was used and concludes with " `Imputation complete.`". Errors (e.g. undefined dataset, column not found) are reported with messages.

2.1.6 Commands: `impute missing multiple`

Purpose Fills in missing values in a specific column by computing group-wise and time-wise averages. This version is intended for datasets with more complex or multiple treatment groups (e.g., staggered adoption or dynamic panels).

Syntax

```
impute missing multiple for "column_name" on dataset_name
```

Arguments

- `"column_name"` — The column where missing values should be filled in.
- `dataset_name` — The name of the dataset, as loaded via `load dataset`.

Required setup Before using this command, the following must be set for the dataset:

- A group column via `set group column`
- A time column via `set time column`
- Optionally, a subgroup column via `set subgroup column` (if finer grouping is desired)

What it does This command:

- Calculates the average value of the chosen column for each group and time combination
- If a subgroup column is defined, it computes even more precise averages using subgroup information
- Replaces missing values in the column with these averages

When to use this Use this version instead of `impute missing data` when:

- Your dataset includes many time periods or rolling treatment cohorts
- You are not working with clearly defined pre/post-treatment splits

Example

```
impute missing multiple for "lemp" on mpdta_prova
```

Notes

- The original dataset is updated in memory with the imputed values
- Only missing values are changed; existing (non-missing) entries remain untouched
- This command is especially useful in event-study or staggered DiD datasets

2.2 Data Inspection

2.2.1 Commands: `compute pre post means`

Purpose This command calculates and displays the average value of an outcome variable in the **pre-treatment** and **post-treatment** periods, separately for each treatment group.

Syntax

```
compute pre post means [for "outcome_column"] on dataset_name
```

Arguments

- `"outcome_column"` (optional) — The column to average. If omitted, the default outcome set earlier will be used.
- `dataset_name` — The name of the dataset (as defined using `load dataset`).
- `na omit` — the command drops any row that has a missing value in any analysis variable before computing.

Required setup Before using this command, you must first:

- Set a group column using `set group column`
- Set a time column with pre- and post-treatment periods using `set time column`
- Optionally, set an outcome column using `set outcome column` (if not specified in the command itself)

What it does For each treatment group in the dataset, this command:

- Computes the average outcome in the pre-treatment period
- Computes the average outcome in the post-treatment period
- Displays the results in a simple table

Supports multiple treatment groups If there are multiple treated groups (e.g., based on different years of treatment), the command computes separate pre/post averages for each one. This allows comparison across treatment cohorts.

Example

```
compute pre post means for "lemp" on mpdta_prova
```

Output example

```
Group      Period           Mean
2004       pre-treatment     10.5
2004       post-treatment    12.3
2006       pre-treatment     11.1
2006       post-treatment    13.7
...
```

Notes

- This command is useful for quick diagnostics and comparing treatment effects visually.
- It does not modify the dataset.
- The results are printed to the screen for inspection.

2.2.2 Commands: `plot distribution by`

Purpose Visualise how the *average* of a chosen variable differs

- across the treatment groups defined with `set group column`, and
- *either*
  - across a user–supplied categorical column, or
  - simply between the **Pre** and **Post** periods that were set with `set time column`.

Syntax

```
plot distribution by "category_column" [for "value_column"] on dataset_name
plot distribution by                   [for "value_column"] on dataset_name
```

Two equivalent signatures therefore exist:

1. **With a category column**
   `plot distribution by "chain" for "emptot" on employment`
2. **Without a category column**
   `plot distribution by for "sales" on stores`

Arguments

`"category_column"` Optional.
: A categorical column to place on the x-axis (e.g. year, region, chain). If omitted, the command defaults to a simple *Pre vs Post* comparison.

`"value_column"` Optional.
: The numeric variable whose mean will be plotted. If omitted the outcome column previously set with `set outcome column` is used.

`dataset_name` Mandatory.
: The short name you provided when you called `load dataset`.

`na omit` — the command drops any row that has a missing value in any analysis variable before computing/plotting/fitting.

Behaviour

- The command computes the mean of `value_column` for every combination of
  - the **group column** (treatment/control), and
  - either the chosen `category_column` *or* the Pre/Post period.
- A grouped bar chart is produced with bars side-by-side for the treatment groups.
- Rows where any of the required columns are missing are silently skipped.

Prerequisites Before calling this command you must have already:

1. Loaded the data with `load dataset`;
2. Declared a group column via `set group column`;
3. Declared a time column (to identify Pre/Post) with `set time column`;
4. Optionally set an outcome with `set outcome column`.

Examples

```
% 1. Compare average employment (emptot) across chains,
%    separately for New Jersey vs Pennsylvania:
plot distribution by "chain" for "emptot" on employment

% 2. Quick pre/post check of average sales using the default outcome column:
plot distribution by on retail_data
```

Typical Questions it Helps Answer

- "Do treated stores outperform controls within each retail chain?"
- "Has the treatment shifted the overall mean outcome from the pre-period to the post-period?"

2.2.3 Commands: `plot eventstudy means`

Purpose Visualises the average difference in outcome between treated and control groups over time, centered around each group's treatment year. This is useful for exploring dynamic treatment effects and checking pre-treatment trends.

Syntax

```
plot eventstudy means [for "outcome_column"] on dataset_name [based on "not-yet-treated" | '
```

Arguments

- `"outcome_column"` (optional) — The outcome variable to plot. If omitted, uses the column set by `set outcome column`.
- `dataset_name` — The dataset to use (must already be loaded).
- `based on "..."` (optional) — Specifies the control group strategy:
  - `"not-yet-treated"` (default): compares each treated group to units not yet treated at that time.
  - `"never-treated"`: compares to units that are never treated (group = 0).

Required Setup Before running this command, the following must be defined:

- A group column using `set group column`
- A time column using `set time column`
- An outcome column using `set outcome column` (or specify it in the command)
- An ID column using `set id column`

What it does

- For each treated group, aligns time so that the treatment year becomes event time = 0.
- Computes and plots the average difference in outcome between treated units and controls at each event time (before and after treatment).
- Produces one chart per treated group.

Example Usage

```
plot eventstudy means on mpdta_prova
plot eventstudy means for "lemp" on mpdta_prova based on "never-treated"
```

Output Each chart displays:

- x-axis: event time (years relative to treatment)
- y-axis: average difference in outcomes between treated and control units
- A vertical dashed line at year 0 to mark the treatment moment

Interpretation

- Flat lines before treatment suggest parallel trends
- Jumps after event time = 0 suggest treatment effects
- You can compare the impact timing across different treatment cohorts

Notes

- Only works with datasets where treatment groups are coded as numeric values (e.g. years)
- Useful for event-study visualisation and exploratory analysis before running formal estimators
- You must define treatment timing using `set group column` with year values

2.3 Assumptions check

2.3.1 Commands: `check parallel trends`

Purpose Tests the **parallel trends assumption** for continuous outcomes in a standard two–group Difference-in-Differences (DiD) setup by checking whether treated and control groups have similar pre-treatment trends.

Syntax

```
check parallel trends [for "outcome_column"]
                      [on dataset_name]
                      [with p-value threshold N]
                      [using robust]
                      [using clustered [by "colA[,colB,...]"]]
                      [na omit]
```

Arguments

- `"outcome_column"` (optional) — Outcome to test; if omitted, uses the column set via `set outcome column`.
- `dataset_name` (optional) — Target dataset; if omitted, uses the last loaded dataset.
- `with p-value threshold N` (optional; default `0.05`) — Significance cut-off for rejecting parallel trends.
- `using robust` (optional) — Use Huber–White HC1 heteroskedasticity-consistent SEs.
- `using clustered [by "..."]` (optional) — Use cluster-robust SEs.
  - If `by` is provided, cluster on the given column(s). Multiple columns may be given as a comma-separated list for multi-way clustering.
  - If `by` is omitted, clusters default to the column set via `set id column`.
- `na omit` (optional; backward-compatible) — Accepted for compatibility, but *redundant*: the command **always** drops rows with NA in any analysis column (see NA handling below).

Required Setup Before running the command, you must:

- Define a treatment group column: `set group column ... with treatment values ...`
- Define a time column and pre/post periods: `set time column ... pre-period ... post-period ...`
- (Optionally) set the outcome: `set outcome column ...`

- (Optional) set covariates: `set covariates "..."` — if set, they are **automatically included** in the test model.
- (Recommended for clustering) set a unit identifier: `set id column ...`

What It Does

1. Detects whether the time column is numeric or date-like; if date-like, parses common formats.
2. Filters to *pre-treatment* observations only; requires at least two distinct pre-periods.
3. Builds:
   - `treatment` = 1 for treated group, 0 otherwise (within pre-periods), and
   - `time_numeric` = a zero-based index of pre-period time.
4. **NA handling**: prior to estimation, drops any row with NA in *any* variable used by the model: the outcome, `treatment`, `time_numeric`, all covariates (if set), and any cluster-by columns (if clustering is requested). A message reports how many rows were dropped.
5. Runs OLS:

   outcome $\sim$ treatment$\times$time_numeric + (all covariates set via `set covariates`, if any)

   with the requested covariance estimator:

   - **Clustering takes precedence** over robust. If both are specified, clustered SEs are used.
   - If clustering is requested but the cluster column(s) are unavailable, it **falls back** to HC1 (if `using robust` was given) or to non-robust SEs.
   - If `using clustered` is given without `by ...`, it clusters on the `id_col` set via `set id column`.
   - If `by` specifies multiple columns, multi-way clustering is applied.

Output

- Printed regression summary (coefficients, standard errors, R^2, etc.), the exact formula used, and the number of rows dropped due to NA.
- The p-value for the interaction term `treatment:time_numeric`.
- Decision based on the threshold:
  - $p <$ threshold: **Parallel trends assumption fails**.
  - $p \geq$ threshold: **Parallel trends assumption holds**.

Interpretation

- **p** < threshold — Evidence of different pre-treatment slopes across groups; the DiD identifying assumption may be violated.
- **p** ≥ threshold — No evidence against parallel trends in the pre-period.

Examples

```
check parallel trends on employment using robust
check parallel trends for "lemp" on mpdta_prova with p-value threshold 0.10
check parallel trends on standard_did using clustered
check parallel trends on standard_did using clustered by "state"
# Covariates are used automatically if previously set:
set covariates "age,income" on multiclass
check parallel trends on multiclass using clustered by "id"
```

Notes & Fallbacks

- Works with numeric or date-string time columns; several date formats are tried automatically.
- Requires at least two distinct pre-treatment periods and variation in treatment within pre-period data.
- If clustering is requested but no valid cluster columns are available:
  - Falls back to HC1 if `using robust` was also specified; otherwise uses non-robust SEs.
  - A clear warning is printed describing the fallback.
- **Covariates**: if you called `set covariates "..."` earlier, those covariates are automatically added to the model—no extra flag is needed.
- **NA handling**: rows with missing values in any analysis variable (outcome, `treatment`, `time_numeric`, covariates, cluster-by columns) are always dropped so the model and covariance estimators align. The `na omit` token is accepted for backward compatibility but is not required.

2.3.2 Commands: `check parallel trends staggered`

Purpose Tests the validity of the **parallel trends assumption** in multiple–treatment (staggered adoption) settings, a prerequisite for causal interpretation in Difference-in-Differences (DiD) and event-study designs. It checks whether treated and control cohorts evolved similarly *before* treatment.

Syntax

```
check parallel trends staggered [for "outcome_column"]
                                on dataset_name
                                [based on "method"]
                                [with p-value threshold N]
                                [using robust]
                                [using clustered]
                                [using clustered by "colA[,colB]"] [na omit]
```

Arguments

- `"outcome_column"` (optional) — Outcome to test. If omitted, uses the one set via `set outcome column`.
- `dataset_name` (required) — A dataset previously loaded.
- `based on "..."` (optional) — Choice of control strategy:
  - `"interaction"` (default): pre-treatment regression with a time trend interacted with treatment.
  - `"not-yet-treated"`: compare each treated cohort to units that will be treated in the future (ideal for staggered adoption).
  - `"never-treated"`: compare to units that are never treated (group = 0).
- `with p-value threshold N` (optional) — Significance cutoff; default `0.05`.
- `using robust` (optional) — Use HC1 (Huber–White) heteroskedasticity-consistent SEs (no clustering).
- `using clustered` (optional) — Cluster-robust SEs using the dataset's `id_col` (must be set via `set id column ...`).
- `using clustered by "colA[,colB]"` (optional) — Cluster on specific column(s). If two columns are given (e.g., `"state,year"`), two-way clustering is applied. If the requested columns are missing, the command falls back to robust or classical SEs with a clear message.
- `na omit` Optional. The command drops any row that has a missing value in any analysis variable before computing/plotting/fitting.

Required Setup

- `set group column ...` (e.g., cohort/treatment timing indicator).
- `set time column ...` (time variable; numeric or parseable date).
- `set outcome column ...` (recommended; otherwise specify `for "..."}` in the command).

- `set id column ...` is required *only* if you intend to use `using clustered` without `by "..."` (so the function knows which unit ID to cluster on).
- Optional: `set covariates "x1,x2,..."` to include controls in the pre-trend regression.

What it does

- Constructs a pre-treatment sample according to the chosen method and, for each cohort (when applicable), runs:

  `outcome` ~ `treatment` + `time_numeric` + `treatment:time_numeric` [+ covariates]

- Reports the p-value for the interaction term `treatment:time_numeric`; this tests whether treated and control units have equal trends prior to treatment.
- Uses the requested variance estimator: classical OLS, HC1 robust, or cluster-robust (one- or two-way).
- **Missing data handling:** rows with missing values in any model variable (outcome, treatment, time_numeric, covariates) and any requested cluster column(s) are dropped *before* fitting so the estimation sample and clustering groups align. This mirrors `statsmodels`' internal behavior and prevents length mismatches.
- Cohorts with insufficient pre-period variation (e.g., fewer than two distinct time points or no treated/control variation) are skipped with a notice.

Control Group Methods

- `interaction` — Standard pre-trend test using a time trend interacted with treatment, run per cohort (staggered) or once (single-treatment).
- `not-yet-treated` — Compares each cohort to units that will be treated later (recommended in staggered adoption).
- `never-treated` — Compares to never-treated units (`group = 0`) in the pre-period.

Examples

```
check parallel trends staggered for "lemp" on mpdta_prova

check parallel trends staggered on mpdta_prova
  based on "never-treated"
  with p-value threshold 0.1

check parallel trends staggered on mpdta_prova
```

```
  based on "not-yet-treated"
  using robust

check parallel trends staggered on mpdta_prova
  based on "never-treated"
  using clustered                % clusters by id_col

check parallel trends staggered on mpdta_prova
  based on "never-treated"
  using clustered by "countyreal" % explicit 1-way clustering

check parallel trends staggered on mpdta_prova
  based on "interaction"
  using clustered by "state,year" % 2-way clustering
```

Output

- For each applicable cohort/method, prints the regression summary and the p-value on `treatment:time_numeric`.
- If `p < threshold`: warns that parallel trends may not hold.
- If `p  threshold`: indicates parallel trends are plausible.

Interpretation

- The coefficient on `treatment:time_numeric` captures pre-treatment trend differences between treated and control units.
- Choice of SEs (classical vs HC1 vs clustered) affects the p-value; with panels, clustered SEs are typically preferred.

Notes

- Works for single-treatment and staggered multi-cohort designs.
- Includes any covariates set via `set covariates ...`.
- Automatically skips cohorts with insufficient pre-period data.
- When `using clustered` without `by "..."` the `id_col` must be set; otherwise the command falls back to robust or classical SEs with an explanatory message.

2.3.3 Commands: `check parallel trends multiclass`

Purpose
Tests the **parallel trends assumption** when the *outcome variable is categorical or takes on multiple discrete values*. Instead of relying on pre-treatment *trend slopes*, this command compares how the outcome transitions between discrete states, separately for treated and control groups, using a chi-squared test.

Syntax

```
check parallel trends multiclass [for "outcome_column"]
                                 on dataset_name
                                 [with p-value threshold 0.05] [na omit]
```

Arguments

- `"outcome_column"` (optional): The name of the outcome variable to test. If omitted, the column set via `set outcome column` is used.
- `dataset_name`: The name of the dataset previously loaded using `load dataset`.
- `with p-value threshold N` (optional): Sets the significance threshold for the test (default is `0.05`).
- `na omit` Optional. The command drops any row that has a missing value in any analysis variable before computing/plotting/fitting.

Required Setup Before using this command, you must:

- Define a group column using `set group column`. The treated group must be coded as `1`; all other values are treated as control.
- Define a time column using `set time column`. Time can be numeric (e.g., `-1`, `0`, `1`) or years (2003, 2004) or strings like `"Feb-92"`.
- Optionally define the outcome column via `set outcome column`.

What It Does

1. Converts the time column to a numeric format if needed (e.g., parses `"Feb-92"` into year-month integer).
2. Filters the data to pre-treatment periods only (`time <= 0`).
3. For each group (treated/control), identifies the earliest and latest pre-treatment time points.
4. Computes outcome **transition matrices** from those two time points.
5. Compares the treated vs control transition probabilities.
6. Uses a chi-squared test on raw counts to assess whether transition patterns differ significantly.

Output

- Prints each group's transition matrix (pre → post outcome).
- Displays a table of differences between treated and control group probabilities.
- Outputs the chi-squared test p-value and states whether parallel trends hold under the threshold.

Interpretation

- If `p < threshold` ⇒ Parallel trends assumption may be violated.
- If `p >= threshold` ⇒ Parallel trends assumption appears valid.

Example Usage

```
check parallel trends multiclass on multiclass
check parallel trends multiclass for "outcome" on multiclass with p-value threshold 0.1
```

When to Use

- When the outcome is categorical (e.g., grade levels, satisfaction tiers, health statuses).
- When you want to assess the similarity of outcome dynamics between groups before treatment.

Notes

- Requires at least two distinct pre-treatment time points.
- The column `case` must exist and uniquely identify units across time (used to compute transitions).
- Only applies to discrete (non-continuous) outcomes.

2.3.4 Commands: `check parallel trends count binary`

Purpose Tests the **parallel trends assumption** for count or binary outcome variables using a regression model. This check helps determine whether treated and control groups followed similar patterns before treatment — a key condition for causal inference using difference-in-differences (DiD).

Syntax

```
check parallel trends count binary [for "outcome_column"]
on dataset_name
[using distribution "poisson" | "lpm"]
[with p-value threshold 0.05]
[using robust]
[using clustered]
[using clustered by "cluster_col"] [na omit]
```

Arguments

- `"outcome_column"` *(optional)* : *The variable to test. If omitted, the outcome previously set with* `set outcome colu`

Required Setup Before using this command, you must:

- Declare a treatment group using `set group column`
- Declare an id column using `set id column`
- Declare time information using `set time column`, including `pre-period` values
- Optionally, set the outcome column using `set outcome column`

What It Does

- Converts the time column to numeric if needed (e.g., from `"Feb-92"` to an internal number scale)
- Keeps only the rows that fall in the pre-treatment period
- Identifies treated and control units using your treatment group setup
- Runs a regression model of the form:

  outcome treatment × time outcome treatment × time
- Checks whether there is a statistically significant difference in pre-treatment trends
- Displays a model summary and a pass/fail message based on the p-value of the interaction term

Time Format Support This command accepts various formats in the time column:

- Calendar years: `2003`, `2004`, etc.
- Event time: `-1`, `0`, `1`, etc.
- Month strings: `"Feb-92"`, `"Jan-2020"`, etc.

If the time column is in string format, it will be automatically converted to a numeric scale internally.

Interpretation

- If the p-value of the treatment-time interaction is **below** the threshold: pre-trends may differ → assumption fails
- If the p-value is **above** the threshold: pre-trends are statistically similar → assumption holds

Example Usage

```
check parallel trends count binary on count

check parallel trends count binary for "tot_notechs" on count
using distribution "poisson"

check parallel trends count binary on accidents
with p-value threshold 0.1 using robust

check parallel trends count binary on count using clustered
check parallel trends count binary on count using clustered by "hospital_id"
```

Notes

- Choose `"poisson"` for count data (e.g., number of crimes, visits, etc.)
- Use `"lpm"` for binary outcomes (e.g., treated or not, success/failure)
- You must provide at least two distinct time values in the pre-period
- If the time column cannot be interpreted (e.g., messy text), an error is printed and the test is skipped
- Clustering behavior: without `by`, the ID column is used; with `by`, the named column is used. If the requested cluster column is unavailable, the command falls back to the ID column (if set), otherwise to robust/non-robust SEs, printing a warning.

2.4 Running Diff-in-Diff

2.4.1 Commands: `run did regression`

Purpose Runs a classical Difference-in-Differences (DiD) regression using a linear model with an interaction between treatment status and the post–treatment indicator. Optionally includes previously defined covariates, supports robust or cluster–robust standard errors, and can drop rows with missing values in the analysis variables.

Syntax

```
run did regression [for "outcome_column"]
                   [on dataset_name]
                   [using covariates]
                   [using robust]
                   [using clustered [by "colA[,colB,...]"]]
                   [na omit]
```

Arguments

- `"outcome_column"` (optional) — Outcome to analyze; if omitted, uses the column set via `set outcome column`.
- `dataset_name` (optional) — Target dataset; if omitted, uses the last loaded dataset.
- `using covariates` (optional) — Adds all covariates previously defined with `set covariates`.
- `using robust` (optional) — Use Huber–White HC1 heteroskedasticity–consistent SEs.
- `using clustered [by "..."]` (optional) — Use cluster–robust SEs.
  - If `by` is provided, cluster on the specified column(s). Multiple columns may be supplied as a comma–separated list.
  - If `by` is omitted, clusters default to the column set via `set id column`.
- `na omit` (optional) — Before fitting, **drops any row that has an NA in any analysis variable**: the outcome, `treatment`, `post`, `interaction`, any included covariates (if `using covariates` is present), and any cluster-by columns (or the `id_col` if clustering without `by ...`). If too few rows remain, the command aborts with a clear message.

Required prior configuration

- `load dataset ... as NAME`
- `set group column "..."` *with treatment values* `...`
- `set time column "..."` *pre-period* `...` *post-period* `...`
- (Optional) `set outcome column "..."`
- (Optional) `set covariates "..."`
- (Recommended for clustering) `set id column "..."`

What it does

1. Constructs a binary `post` indicator from the configured pre/post periods (rows not in either set are dropped).
2. Builds `treatment` = 1 for units in the configured treated group, 0 otherwise.
3. Forms the interaction `interaction = treatment` × `post`.
4. If `na omit` is specified, removes all rows with missing values in any analysis variable (outcome, `treatment`, `post`, `interaction`, selected covariates, and requested cluster-by columns). Reports how many rows were dropped; aborts if the remaining sample is insufficient.
5. Fits an OLS model:

$$\text{outcome} \sim \text{treatment} + \text{post} + \text{interaction} \; [+\text{covariates}],$$

   where the coefficient on `interaction` is the DiD effect.
6. Applies the requested covariance estimator:
   - **Clustering takes precedence** over robust. If both are specified, clustered SEs are used.
   - If clustering is requested but no valid cluster column(s) are available, it **falls back** to HC1 (if `using robust` is present) or to non-robust SEs, and prints a clear warning.
   - If `using clustered` is given without `by ...`, clustering defaults to the `id_col` set with `set id column`.
   - If multiple columns are passed in `by`, multi-way clustering is applied.
7. Stores the DiD estimate and p-value internally (keys: `DID_EFFECT`, `DID_PVALUE`) and prints the full model summary.

Output

- Full regression summary (coefficients, standard errors, R^2, etc.).
- The DiD estimate (coefficient on `interaction`) and its p-value, also saved to global results for reuse.

Example usage

```
run did regression on employment
run did regression for "sales" on retail_data using covariates
run did regression on panel_data using clustered
run did regression on panel_data using clustered by "firm_id,year"
run did regression on survey using robust using covariates
run did regression on employment using clustered by "state" na omit
```

Notes

- If no outcome is given and none has been set, the command aborts with an error.
- Time values are matched to pre/post periods as strings when needed; ensure consistent formatting.
- The treatment effect is the coefficient on `interaction`.
- When clustering is requested, defining a valid `id_col` via `set id column` is recommended unless `by "..."` is specified.
- Without `na omit`, the command does not proactively clean missing values; using `na omit` ensures a consistent estimation sample across covariates and cluster variables.

2.4.2 Commands: `run did fixed effects`

Purpose Estimate a Difference-in-Differences (DiD) model with unit and time fixed effects, allowing for the control of unobserved heterogeneity across units and over time. Including robust standard errors helps to obtain consistent inference in the presence of heteroskedasticity or within-cluster correlation.

Syntax

```
run did fixed effects [for "outcome_column"] on
dataset_name [using covariates] [using robust] [na omit]
```

Arguments

`"outcome_column"` Optional.
: The outcome variable to be analyzed. If omitted, uses the column set via `set outcome column`.

`dataset_name` Mandatory.
: The name of the dataset previously loaded with `load dataset`.

`using covariates` Optional.
: Include additional covariates (previously set via `set covariates`) in the regression.

`using robust` Optional. In the fixed-effects regression, `using_robust` computes heteroskedasticity-consistent (HC1, Huber–White) standard errors without referring to the `id_col`, whereas `using_clustered` computes cluster-robust standard errors by clustering on the `id_col` (which also appears as the unit fixed effect in the formula); if neither flag is specified, ordinary OLS standard errors assuming homoskedasticity and no within-cluster correlation are used.

`na omit` Optional. The command drops any row that has a missing value in any analysis variable before computing/plotting/fitting.

Prerequisites Before running this command, you must have:

- Loaded the dataset using `load dataset`.
- Set the group (treatment) column via `set group column`.
- Set the time column via `set time column` with clearly defined pre- and post-treatment periods.
- Set the outcome column via `set outcome column`.
- Set the unit identifier column via `set id column` (required for fixed effects and robust errors).
- Optionally set covariates via `set covariates`.

Behavior

- Constructs a regression model including:
  - Treatment indicator,
  - Post-treatment period indicator,
  - Interaction term (treatment × post),
  - Unit fixed effects (via `C(unit_id)`),
  - Time fixed effects (via `C(time_column)`),
  - Optional covariates.
- Fits an OLS regression on the specified dataset.
- If `using robust` is specified, calculates cluster-robust standard errors clustered by the unit identifier.
- Reports the estimated DiD effect coefficient (the interaction term), its p-value, and significance at the 5% level.

Example

```
run did fixed effects for "emptot" on employment using covariates using robust
```

Interpretation

- The coefficient on the interaction term estimates the causal effect of treatment.
- Robust standard errors adjust inference for potential heteroskedasticity and intra-unit correlation.
- Fixed effects control for time-invariant unit characteristics and common shocks across time.

2.4.3 Commands: `run att estimations`

Purpose Estimates group-time average treatment effects (ATT(g, t)) in panel data with staggered treatment adoption. Supports multiple estimation methods including doubly robust (DR), inverse probability weighting (IPW), outcome regression (OR), and a simple group-treatment difference (GT) estimator. Allows flexible control group selection, covariate adjustment, exclusion of pre-treatment periods, and inference with confidence intervals and significance testing.

Syntax

```
run att estimations [for "outcome_column"] on dataset_name
                    [using method "DR" | "IPW" | "OR" | "GT"]
                    [with p-value threshold 0.05]
                    [with alpha 0.05]
                    [using covariates]
                    [exclude pre-treatment]
                    [include confidence intervals]
                    [using robust]
                    [using clustered]
                                [using clustered by "colA[,colB]"
                    [using control "never-treated" | "not-yet-treated"] [na omit]
```

Arguments

- `"outcome_column"` (optional) — The outcome variable. Defaults to the column set by `set outcome column`.
- `dataset_name` — The dataset to analyze.
- `using method` (optional) — Estimation method:
  - `DR` — Doubly Robust estimator.
  - `IPW` — Inverse Probability Weighting.
  - `OR` — Outcome Regression.
  - `GT` — Simple group-treatment difference.
- `with p-value threshold` (optional) — Threshold for determining statistical significance in diagnostics (default: 0.05).
- `with alpha` (optional) — Significance level used for confidence intervals (default: 0.05, yielding 95% intervals).
- `using covariates` (optional) — Adjusts for covariates defined with `set covariates` to improve precision.
- `exclude pre-treatment` (optional) — Omits comparisons involving pre-treatment periods when computing ATT.

- `include confidence intervals` (optional) — Computes and reports confidence intervals for ATT estimates.
- `using robust` (optional) — Uses robust (heteroskedasticity-consistent) standard errors in IPW/OR/GT and in auxiliary models.
- `using clustered` (optional) — Cluster-robust SEs using the dataset's `id_col` (must be set via `set id column ...`).
- `using clustered by "colA[,colB]"` (optional) — Cluster on specific column(s). If two columns are given (e.g., `"state,year"`), two-way clustering is applied. If the requested columns are missing, the command falls back to robust or classical SEs with a clear message.
- `using control` (optional) — Control group selection:
  - `never-treated` — Controls are units never treated (group = 0).
  - `not-yet-treated` — Controls are units not yet treated at time t (default).
- `na omit` Optional. The command drops any row that has a missing value in any analysis variable before computing/plotting/fitting.

Required prior configuration

- `set group column` with treatment cohort identifiers.
- `set time column` with ordered time periods.
- `set outcome column` or specify outcome in the command.
- `set id column` for panel identification.
- `set covariates` if `using covariates` is specified.

Operation For each treated group g and time period t, the command:

- Constructs the appropriate pre/post comparison depending on t relative to g, respecting the `exclude pre-treatment` flag.
- Merges pre- and post-period observations on the panel identifier and computes the outcome change ΔY_i.
- Defines treatment indicator $G_{g,i}$ based on group membership and filters the control set according to `using control`.
- Estimates propensity scores and outcome regression models (if covariates are used), applying robust or clustered corrections if requested.
- Computes ATT using the selected method:

  - `GT`: simple difference in ΔY between treated and control.
  - `IPW`: weighted comparison using estimated propensity scores.
  - `OR`: comparison of fitted outcomes.
  - `DR`: combines outcome regression and weighting with an empirical inference procedure.

- Computes standard errors and forms confidence intervals (normal approximation for IPW/OR/GT; empirical variance approximation for DR) and p-values, marking significance based on the threshold.

Example

```
set group column "first.treat" on mpdta_prova with treatment values "2004,2006,2007"
set time column "year" on mpdta_prova pre-period "2003" post-period "2004,2006,2007"
set outcome column "lemp" on mpdta_prova
set id column "countyreal" on mpdta_prova
set covariates "lpop" on mpdta_prova

run att estimations for "lemp" on mpdta_prova
  using method "GT"
  with p-value threshold 0.1
  with alpha 0.1
  using covariates
  exclude pre-treatment
  using clustered
  using control "not-yet-treated"
```

Notes

- `exclude pre-treatment` skips ATT comparisons where $t < g$ if pre-period effects are not desired.
- Covariate adjustment (`using covariates`) and control group choice affect identification and precision; `not-yet-treated` is typically preferred in staggered settings.
- Robust and clustered options adjust inference to account for heteroskedasticity or within-unit correlation.
- The `with alpha` option controls the width of confidence intervals (e.g., $\alpha = 0.1$ yields 90% intervals).

2.4.4 Commands: `run did multiclass`

Purpose Performs a multinomial Difference-in-Differences regression analysis for datasets with multiple treatment groups or classes. This method models the outcome as a function of group membership, time period (pre/post), and their

interaction, optionally adjusting for covariates. It is suitable for settings where treatment status is categorical with more than two groups.

Syntax

```
run did multiclass [for "outcome_column"] on dataset_name
  [using covariates] [using robust]
  [using clustered] [using clustered by "colA,colB"] [na omit]
```

Arguments

- `"outcome_column"` (optional) — The name of the outcome variable to analyze. If omitted, the previously set outcome column on the dataset is used.
- `dataset_name` — The alias of the dataset as loaded via `load dataset`.
- `using covariates` (optional) — If specified, includes all covariates set on the dataset in the regression.
- `using robust` (optional) — If specified, fits the model using robust (Huber-White) standard errors.
- `using clustered` / `using clustered by "colA,colB"` (optional) — Use cluster-robust standard errors for the multinomial logit. If `by` is omitted, the model clusters by the dataset's `id_col`. If multiple columns are provided, only the *first* existing column is used (MNLogit supports one-way clustering). If the requested column(s) are not found and no `id_col` is available, the command falls back to `HC1` (when `using robust`) or non-robust SEs, with a warning.
- `na omit` Optional. The command drops any row that has a missing value in any analysis variable before computing/plotting/fitting.

Required Setup Before running this command, you must have already:

- Loaded the dataset with `load dataset`.
- Set the group column via `set group column` (defining treatment groups).
- Set the time column via `set time column` with pre- and post-treatment period values.
- Set an outcome column via `set outcome column` or specify it in the command.
- Optionally set covariates via `set covariates`.

- **If you plan to use `using clustered` without `by`, set an id via `set id column`.**

Details

- The command converts the time column values to strings and determines which periods are pre- or post-treatment based on the configured period values.
- It creates an indicator variable `post` equal to 1 if the observation is in a post-treatment period, 0 otherwise.
- The interaction term between the group identifier and the post indicator captures the treatment effect.
- When `using covariates` is specified, all covariates set on the dataset are added as controls.
- The multinomial logistic regression model is fitted to estimate treatment effects across multiple classes.
- If `using robust` is specified, robust standard errors are used.
- If `using clustered` is specified, the model is refit with `cov_type='cluster'`:
  - With `by "colA,colB"`: clusters on the first provided column that exists in the data (one-way clustering).
  - Without `by`: clusters on `id_col`.
  - If no valid clustering column is available: falls back to `HC1` (if robust) or non-robust and prints a warning.

Output

- Prints the full regression summary table.
- For each treatment class, reports whether the treatment effect (interaction term) is statistically significant at the configured p-value threshold (default 0.05).
- If the interaction term is missing, a warning is printed.
- When clustering is requested but cannot be applied, a warning explains the fallback (e.g., to `HC1`).

Example

```
run did multiclass for "sales" on retail_data
  using covariates using robust using clustered

run did multiclass for "sales" on retail_data
  using clustered by "state"
```

The first example adjusts for covariates, uses robust SEs, and clusters by `id_col`. The second clusters by the `state` column (one-way clustering).

Notes

- Time values are treated as strings, so both numbers (-1,0,1), numeric years (e.g., "2003") and string periods (e.g., "Feb-92") are supported.
- Make sure pre- and post-period values exactly match the string format of your time column.
- Multinomial DiD allows treatment effects to vary across multiple groups, beyond simple treated/control dichotomies.
- Clustered SEs are *one-way* for this model; if multiple columns are provided after `by`, only the first existing column is used.

2.4.5 Commands: `run did count binary`

Purpose Performs Difference-in-Differences (DiD) analysis for count data with binary treatment status. Supports Poisson regression or Linear Probability Model (LPM) to estimate treatment effects, optionally adjusting for covariates.

Syntax

```
run did count binary [for "outcome_column"] on dataset_name
                     [with p-value threshold 0.05]
                     [using distribution "poisson" | "lpm"]
                     [using robust standard errors]
                     [using covariates]
                     [using bootstrap confidence intervals with n N] [na omit]
```

Arguments

- `"outcome_column"` (optional) — Outcome variable to analyze.
- `dataset_name` — Name of the dataset loaded via `load dataset`.
- `with p-value threshold` (optional) — Threshold for statistical significance (default 0.05).

- `using distribution` (optional) — Model type: `poisson` (default),`lpm` or `logit`.
- `using robust standard errors` (optional) — Enables heteroskedasticity-robust standard errors.
- `using covariates` (optional) — Includes user-defined covariates (previously set) in regression.
- `using bootstrap confidence intervals with n N` (optional) — Computes bootstrap confidence intervals with `N` resamples. If `N` is omitted, defaults to 100 resamples.
- `na omit` (optional) — omitting NAs.

Time Column Handling

- Supports numeric time formats (e.g., `-1`, `0`, `1`, `2003`) directly.
- Attempts to parse non-numeric time columns using the `%b-%y` date format (e.g., `Feb-92`).
- Fails gracefully with an error if the time column is neither numeric nor parseable as dates.
- Determines pre- and post-treatment periods by matching the time column values to configured `pre_period_values` and `post_period_values`.

Example

```
run did count binary for "num_cases" on health_data
run did count binary for "num_visits" on clinic_data using covariates
with p-value threshold 0.1

run did count binary for "admissions" on hospital_data using distribution "lpm"
using robust standard errors

run did count binary for "calls" on support_data
using bootstrap confidence intervals with n 200
```

Output

- Prints model regression summary and key statistics.
- Displays p-value and significance status of the DiD treatment effect.
- If bootstrap enabled, prints 95% confidence intervals for the interaction effect.
- Returns a dictionary containing the regression summary, formula, interaction p-value, and optionally bootstrap CIs.

Notes

- Covariates must be specified beforehand with `set covariates` command.
- The dataset must have treatment group and time columns properly configured.
- Robust standard errors help adjust for heteroskedasticity.

2.4.6 Commands: `check parallel trends count binary staggered`

Purpose Assesses the parallel-trends assumption for count/binary outcomes in a staggered-adoption DiD setup by fitting, for each cohort, a generalized linear model (Poisson, LPM, or Logit) of the outcome on treatment, time, and their interaction, using only pre-treatment observations.

Syntax

```
check parallel trends count binary staggered
    [for "outcome_column"]
    on dataset_name
    [based on "interaction" | "never-treated" | "not-yet-treated"]
    [with p-value threshold N]
    [using distribution "poisson" | "lpm" | "logit"]
    [using robust]
    [using clustered]
    [using clustered by "col1[,col2]"] [na omit]
```

Arguments

- `"outcome_column"` (optional) — Name of the count or binary outcome; defaults to the column set via `set outcome column`.
- `dataset_name` — Identifier of the dataset to analyze.
- `based on` (optional) — Choice of control logic:
  - `"interaction"` (default): classical interaction–time test on pooled pre-period.
  - `"never-treated"`: compares never-treated units only.
  - `"not-yet-treated"`: uses not-yet-treated as controls for each cohort.
- `with p-value threshold N` (optional) — Significance cutoff for the interaction p-value (default 0.05).
- `using distribution` (optional) — Model family: `"poisson"`, `"lpm"` (linear), or `"logit"` (binary); defaults to `poisson`.

- `using robust` (optional) — Fit with robust standard errors (HC1).
- `using clustered` or `using clustered by "col1[,col2]"` (optional) — Request clustered standard errors. If `by` is omitted, clustering defaults to the configured `id_col`. If multiple columns are listed, only the first available is used (one-way clustering). If requested columns are not found, the routine falls back to `id_col` when available; otherwise it falls back to HC1 (if `using robust`) or non-robust, with a warning.
- `na omit` (optional) — omitting NAs.

Required prior configuration

- Dataset loaded via `load dataset`.
- Treatment group column set via `set group column`.
- Time column and pre/post periods set via `set time column`.
- (Optional) Outcome column set via `set outcome column`.
- (Optional) Covariates defined via `set covariates`.

What it does

1. Converts the time column to numeric and computes event time relative to each cohort's adoption year.
2. For each treated cohort g:
   - Subsets to observations with cohort $= g$ or cohort $> g$.
   - Flags `treatment` $= 1$ for cohort g, 0 otherwise.
   - Keeps pre-treatment rows (`time_numeric` < 0).
   - Skips cohort if fewer than two pre-period time points or no treated/-control variation.
   - Fits the specified GLM/LPM/Logit model: `outcome` $\sim$ `treatment` $+$ `time_numeric` $+$ `treatment` $\times$ `time_numeric`[$+$ covariates].
   - Applies covariance as requested: clustered (if `using clustered`), else HC1 (if `using robust`), else non-robust. One-way clustering only; falls back as described above with warnings.
   - Prints model summary and interaction p-value, indicating whether parallel trends hold.

Example

```
check parallel trends count binary staggered
  on count_data
  based on "not-yet-treated"
  with p-value threshold 0.1
  using distribution "poisson"
  using clustered by "firm_id"
```

Output For each treated cohort, prints the model summary and whether the pre-period interaction term is statistically significant at the specified threshold.

2.4.7 Commands: `run att count binary`

Purpose Estimates group-time average treatment effects ($\text{ATT}(g,t)$) for count or binary outcomes in panel data with staggered treatment adoption. Supports multiple estimation methods (DR, IPW, OR, GT), allows choice of outcome distribution (Poisson, linear probability/LPM, or logit), covariate adjustment, exclusion of pre-treatment comparisons, and flexible control group selection. Reports both raw ATT and log-scale ATT (log ratio of treated to control means), with inference via normal-approximation confidence intervals and p-values.

Syntax

```
run att count binary [for "outcome_column"] on dataset_name
                     [using method "DR" | "IPW" | "OR" | "GT"]
                     [with p-value threshold 0.05]
                     [using distribution "poisson" | "lpm" | "logit"]
                     [using covariates]
                     [exclude pre-treatment]
                     [using control "never-treated" | "not-yet-treated"]
                     [using clustered [by "column_name"]] [na omit]
```

Arguments

- `"outcome_column"` (optional) — The outcome variable (count or binary). Falls back to the column set by `set outcome column`.
- `dataset_name` — The dataset to analyze.
- `using method` (optional) — ATT estimator:
  - `DR` — Doubly robust combination of outcome regression and weighting.
  - `IPW` — Inverse probability weighting.
  - `OR` — Outcome regression.
  - `GT` — Simple group-treatment difference on the change in outcome.

- `with p-value threshold` (optional) — Significance cutoff for assessing statistical significance (default: 0.05).
- `using distribution` (optional) — Model for the underlying outcome in auxiliary fits:
  - `poisson` — Poisson specification (default).
  - `lpm` — Linear probability model (ordinary least squares on the differenced outcome).
  - `logit` — Logistic model (requires binary outcome).
- `using covariates` (optional) — Adjusts for covariates defined with `set covariates`.
- `exclude pre-treatment` (optional) — Omits ATT comparisons where $t < g$ (pre-treatment) if not desired.
- `using control` (optional) — Control group choice:
  - `never-treated` — Uses never-treated units (group = 0) as controls.
  - `not-yet-treated` — Uses units not yet treated at time t (default).
- `using clustered [by "column_name"]` (optional) — Requests clustered handling consistent with other commands. If `by` is omitted, the ID column set via `set id column` is used; if the specified column is not found, the command falls back to the ID column when available (otherwise the clustering request is ignored).
- `na omit` (optional) omit NA values.

Required prior configuration

- `set group column` with treatment cohort identifiers.
- `set time column` with ordered time periods.
- `set outcome column` or specify outcome in the command.
- `set id column` for panel identification (required for matching pre/post).
- `set covariates` if `using covariates` is specified.

Operation For each treated cohort g and time t, the command:

- Skips comparisons with $t < g$ if `exclude pre-treatment` is active; otherwise constructs the appropriate pre/post pair (baseline is $g - 1$ for post periods, previous time for pre periods).
- Merges pre- and post-period observations by panel ID and computes the outcome change ΔY_i.

- Defines treatment indicator $G_{g,i}$ and filters the control set according to the control specification.
- Optionally fits auxiliary models (propensity score and outcome regression) incorporating covariates and the specified distribution.
- Computes:
  - **Raw ATT:** difference between treated and control in ΔY or appropriate fitted values.
  - **Log-scale ATT:** $\log\big(\mu_{\text{treated}}/\mu_{\text{control}}\big)$, where μ are group means.
- Forms standard errors via normal approximation using sample variances of treated and control changes, constructs confidence intervals, computes p-values, and flags statistical significance based on the threshold.

Example

```
set group column "cohort" on sim_data with treatment values "3,5,7"
set time column "time" on sim_data pre-period "1,2,3" post-period "4,5,6,7,8,9,10"
set outcome column "y" on sim_data
set id column "id" on sim_data

run att count binary for "y" on sim_data
    using method "DR"
    using distribution "poisson"
    with p-value threshold 0.05
    using covariates
    exclude pre-treatment
    using control "not-yet-treated"
```

Notes

- Both raw and log-scale ATTs are reported; the log-scale ATT aids interpretation when effects are multiplicative.
- Covariate adjustment and control selection affect identification and precision; `not-yet-treated` is typically preferred in staggered designs.
- Distribution choice governs the form of auxiliary fits (e.g., logistic for binary outcomes or Poisson for counts).
- Confidence intervals and p-values rely on normal approximations using estimated variances from treated and control groups.

Example DSL Script for Staggered Treatment and ATT Estimation

```
load dataset "mpdta_prova.csv" as mpdta_prova
set group column "first.treat" on mpdta_prova with treatment values "2004,2006,2007"
set time column "year" on mpdta_prova pre-period "2003" post-period "2004,2006,2007"
set outcome column "lemp" on mpdta_prova
set covariates "lpop" on mpdta_prova
show dataset mpdta_prova all
impute missing multiple for "lemp" on mpdta_prova
show dataset mpdta_prova all
compute pre post means on mpdta_prova
plot eventstudy means on mpdta_prova based on "never-treated"

check parallel trends staggered on mpdta_prova based on "interaction"
with p-value threshold 0.1
set id column "countyreal" on mpdta_prova

run att estimations on mpdta_prova based on "not-yet-treated" using method "DR"
with p-value threshold 0.1 using covariates exclude pre-treatment include
confidence intervals
```

- `load dataset "mpdta_prova.csv" as mpdta_prova`
  Loads the CSV file `mpdta_prova.csv` and assigns it the alias `mpdta_prova`.
- `set group column "first.treat" on mpdta_prova with treatment values "2004,2006,2007"`
  Specifies the treatment groups based on the `first.treat` column, marking years 2004, 2006, and 2007 as treated cohorts.
- `set time column "year" on mpdta_prova pre-period "2003" post-period "2004,2006,2007"`
  Defines `year` as the time variable, with 2003 as the pre-treatment period and 2004, 2006, 2007 as post-treatment periods.
- `set outcome column "lemp" on mpdta_prova`
  Sets the outcome variable to `lemp` (e.g., employment levels).
- `set covariates "lpop" on mpdta_prova`
  Specifies `lpop` as a covariate to control for in regression and estimation models.
- `show dataset mpdta_prova all`
  Displays the entire dataset `mpdta_prova` currently in memory.

- `impute missing multiple for "lemp" on mpdta_prova`
  Imputes missing values in `lemp` using group, time, and subgroup means appropriate for staggered adoption data.

- `show dataset mpdta_prova all`
  Displays the dataset again to verify imputation results.

- `compute pre post means on mpdta_prova`
  Calculates average outcomes in pre- and post-treatment periods for each group.

- `plot eventstudy means on mpdta_prova based on "never-treated"`
  Produces event-study plots comparing treated groups to never-treated controls over time.

- `check parallel trends staggered on mpdta_prova based on "interaction" with p-value threshold 0.1`
  Tests the parallel trends assumption using an interaction method, with a p-value threshold of 0.1.

- `set id column "countyreal" on mpdta_prova`
  Assigns `countyreal` as the unique identifier for panel units, necessary for ATT estimation.

- `run att estimations on mpdta_prova based on "not-yet-treated" using method "DR" with p-value threshold 0.1 using covariates exclude pre-treatment include confidence intervals`
  Runs ATT estimation using the Doubly Robust (DR) method, comparing treated groups to not-yet-treated controls, including covariates, excluding pre-treatment periods, and reporting confidence intervals.

Example DSL Script: Multiclass Difference-in-Differences Analysis

```
load dataset "simulated_multiclass_did.csv" as multiclass
set group column "group" on multiclass with treatment values 1
set time column "time" on multiclass pre-period -1 0 post-period 1
set outcome column "outcome" on multiclass
set covariates "age,income" on multiclass
check parallel trends multiclass on multiclass with p-value threshold 0.05

run did multiclass on multiclass with p-value threshold 0.05
using robust using covariates
```

- `load dataset "simulated_multiclass_did.csv" as multiclass`
  Loads the CSV file into memory and assigns it the name `multiclass` for later reference.

- `set group column "group" on multiclass with treatment values 1`
  Specifies the `group` column as the treatment group identifier, with value `1` representing treated units.
- `set time column "time" on multiclass pre-period -1 0 post-period 1`
  Defines the `time` column with pre-treatment periods `-1` and `0`, and post-treatment period `1`.
- `set outcome column "outcome" on multiclass`
  Sets the outcome variable for analysis.
- `set covariates "age,income" on multiclass`
  Declares `age` and `income` as covariates to adjust for in the analysis.
- `check parallel trends multiclass on multiclass with p-value threshold 0.05`
  Performs a statistical test of the parallel trends assumption for multiclass treatments, using a 5% significance level.
- `run did multiclass on multiclass with p-value threshold 0.05 using robust using covariates`
  Runs the Difference-in-Differences regression for multiclass treatment, applying robust standard errors and adjusting for covariates.

Example DSL Script: Count Data DiD Analysis with Poisson Model and Bootstrapping

```
load dataset "did_sim.csv" as count
set outcome column "tot_notechs" on count
set group column "DBT_grp" on count with treatment values 1
set covariates "device" on count
set time column "time" on count pre-period -12 0 post-period 1

check parallel trends count binary on count with p-value threshold 0.05
using distribution "poisson"

run did count binary on count with p-value threshold 0.05 using distribution "poisson"
using covariates using robust using bootstrap n bootstrap 200

compute pre post means on count
```

`load dataset "did_sim.csv" as count`
Loads the CSV file `did_sim.csv` and assigns it the name `count` for future commands.

`set outcome column "tot_notechs" on count`
Specifies the outcome variable `tot_notechs` in the `count` dataset.

`set group column "DBT_grp" on count with treatment values 1`
Sets the treatment group identifier to the `DBT_grp` column and defines the treated group as having value `1`.

`set covariates "device" on count`
Declares `device` as a covariate to be used in adjusted estimation.

`set time column "time" on count pre-period -12 0 post-period 1`
Defines the time variable `time` and specifies that periods `-12` and `0` are pre-treatment, while `1` is post-treatment.

`check parallel trends count binary on count with p-value threshold 0.05 using distribution "poisson"`
Runs a statistical test of the parallel trends assumption on the `count` dataset using a Poisson model, flagging potential violations if the p-value is below 0.05.

`run did count binary on count with p-value threshold 0.05 using distribution "poisson" using covariates using robust using bootstrap n bootstrap 200`
Runs a Difference-in-Differences count model using Poisson regression on the `count` dataset, controlling for covariates, using robust standard errors, and bootstrapping confidence intervals with 200 resamples.

`compute pre post means on count`
Calculates and displays average outcome values in the pre- and post-treatment periods for treated and control groups.

3 Work in progress

3.0.1 Commands: `check strong parallel trends`

Purpose Tests the **strong parallel trends** assumption in a *continuous–treatment* DiD by asking whether, in the *pre-treatment* periods, the outcome trend is independent of the treatment *dose*. This generalises the usual binary pre-trend test to settings where treatment intensity D_{it} varies across units and time.

Syntax

```
check strong parallel trends
  [for "outcome_column"]
  [on dataset_name]
  [using function "linear" | "quadratic"]
  [using covariates]
  [using robust]
```

```
[using clustered [by "colA[,colB,...]"]]
[na omit]
```

Arguments

- `"outcome_column"` (optional) — Outcome to test; defaults to the column set via `set outcome column`.
- `dataset_name` (optional) — Target dataset; defaults to the last loaded dataset.
- `using function` (optional) — Functional form for dose trend in the pre-test:
  - `"linear"` (default): allows D_{it} and $t \times D_{it}$.
  - `"quadratic"`: additionally allows D_{it}^2 and $t \times D_{it}^2$.
- `using covariates` (optional) — Includes all covariates set via `set covariates`, *plus* their interactions with pre-period time.
- `using robust` (optional) — Huber–White HC1 SEs.
- `using clustered [by "..."]` (optional) — Cluster-robust SEs; without `by`, clusters on the `id` set by `set id column`. If unavailable, falls back to HC1 (when `using robust`) or classical SEs.
- `na omit` (optional) — Drops rows with NA in any analysis variable before fitting (outcome, dose, time, covariates, cluster columns).

Required Setup

- `set time column "..."` with *at least two* distinct `pre-period` values.
- `set outcome column "... "` and `set dose column "..."`.
- (Recommended for clustering) `set id column "..."`.
- (Optional) `set covariates "x1,x2,..."`.

What it does Let t index the pre-periods only, re-indexed to a numeric `time_numeric`.

- **Linear form** fits (using only pre data)

$$Y_{it} \;=\; \alpha + \theta\,\texttt{time\_numeric}_t + \phi\,D_{it} + \psi\,\big(\texttt{time\_numeric}_t \times D_{it}\big) + X_i'\delta \;+\; \texttt{time\_numeric}_t \cdot X_i'\kappa + \varepsilon_{it}.$$

- **Quadratic form** adds D_{it}^2 and $\texttt{time\_numeric} \times D_{it}^2$.

It then tests the joint null that the *pre-trend does not vary with dose*:

$$H_0^{\text{linear}}:\ \psi = 0 \qquad \text{or} \qquad H_0^{\text{quad}}:\ \psi = \psi_2 = 0,$$

where ψ_2 is the coefficient on $\texttt{time\_numeric} \times D_{it}^2$.

Output

- The exact formula used and a regression summary.
- A clear decision:
  - $p <$ threshold $\Rightarrow$ Fails: evidence that pre-trends vary with dose.
  - $p \geq$ threshold $\Rightarrow$ Holds: no evidence against strong PT.
- If fewer than two distinct pre periods are detected, prints:
  `Strong-PT check needs at least two distinct pre-treatment periods.`

Example

```
load dataset "two_period_continuous_sim.csv" as cont_strong
set id column "id" on cont_strong
set time column "time" on cont_strong pre-period -1 0 post-period 1
set outcome column "y" on cont_strong
set dose column "dose" on cont_strong
set covariates "x1,x2" on cont_strong

check strong parallel trends on cont_strong
  using function "linear" using covariates using robust na omit
```

Notes

- This test is *specific* to continuous treatment: with binary treatment, the standard group×time pre-trend test suffices.
- Including `time` $\times X$ lets observed composition X drift differently over time without falsely flagging dose-trend differences.

3.0.2 Commands: `run did continuous`

Purpose Estimate DiD with a *continuous* post-period dose. The command reports:

1. a **binary contrast at the origin** ("any positive post dose" vs. "zero post dose"),
2. a **global** per-unit effect (slope) of the change in dose on the change in outcome, and
3. optionally, a **local** "nudge/ACR" effect at a chosen post-period dose level d via local weighting.

Syntax

```
run did continuous
  [for "outcome_column"]
  [on dataset_name]
  [using covariates]
  [using robust]
  [using clustered [by "colA[,colB,...]"]]
  [at level d] [with bandwidth h]
  [na omit]
```

Arguments

- `"outcome_column"` (optional) — Defaults to the previously set outcome.
- `dataset_name` (optional) — Defaults to the last loaded dataset.
- `using covariates` (optional) — Adds configured covariates (for two-period fits, numeric covariates are used via post–pre differences when available).
- `using robust` / `using clustered [by "..."]` — Inference options; if both are set, clustered SEs take precedence. If `by` is omitted, the ID column is used (when available).
- `at level d`, `with bandwidth h` (optional) — Adds the *local* ACR at post-dose level d using kernel weights with bandwidth h.
- `na omit` (optional) — Drops rows with NA in any analysis or clustering variable before fitting.

Required Setup

- `set time column "..."` with one or more `pre-period` values and at least one `post-period` value.
- `set outcome column "... "` and `set dose column "..."`.
- (Recommended for clustering) `set id column "..."`.
- (Optional) `set covariates "x1,x2,..."`.

What it does Let the last configured pre period be $t = 0$ and the first configured post period be $t = 1$. Keep units observed in both periods and form first differences for each unit i:

$$\Delta Y_i \equiv Y_{i1} - Y_{i0}, \qquad \Delta D_i \equiv D_{i1} - D_{i0}, \qquad \Delta X_{ik} \equiv X_{ik,1} - X_{ik,0}.$$

1. **Binary contrast at the origin (reported by default):** Define $T_{i1} = \mathbf{1}\{D_{i1} > 0\}$ and estimate by OLS

$$\Delta Y_i \;=\; \alpha \;+\; \beta\, T_{i1} \;+\; \Delta X_i'\gamma \;+\; u_i.$$

   $\widehat{\beta}$ is the mean difference in ΔY between units with positive post dose and units with zero post dose.

2. **Global slope (reported by default):** OLS of

$$\Delta Y_i \;=\; \alpha \;+\; \tau\, \Delta D_i \;+\; \Delta X_i'\gamma \;+\; u_i,$$

   where $\widehat{\tau}$ summarizes the average change in ΔY per one-unit change in ΔD.

3. **Local ACR at level d (optional):** On units with $D_{i1} > 0$, run a locally weighted change regression centered at d:

$$\Delta Y_i \;=\; \alpha(d) \;+\; \tau(d)\,(D_{i1} - d) \;+\; \Delta X_i'\gamma(d) \;+\; u_i,$$

   with Epanechnikov weights

$$w_i(d;h) \;\propto\; 0.75\big(1 - u_i^2\big)\,\mathbf{1}\{|u_i| \le 1\}, \quad u_i = \tfrac{D_{i1}-d}{h}, \quad \sum_i w_i(d;h) = 1.$$

   For small h, $\widehat{\tau}(d)$ approximates the average causal response (ACR)—the marginal effect of nudging the post-period dose around level d.

Robust or cluster-robust SEs are applied as requested. If clustering columns are missing, the procedure falls back safely to robust/non-robust SEs.

Output

- **Binary contrast:** $\widehat{\beta}$, its standard error and p-value, plus a regression summary.
- **Global:** $\widehat{\tau}$, its standard error and p-value, plus a regression summary.
- **Local (if requested):** $\widehat{\tau}(d)$ and (when identifiable) $\widehat{\alpha}(d)$, with the chosen h and inference.

Example

```
load dataset "two_period_continuous_sim.csv" as cont2
set id column "id" on cont2
set time column "time" on cont2 pre-period 0 post-period 1
set outcome column "y" on cont2
set dose column "dose" on cont2
set covariates "x1,x2" on cont2

# Global slope + binary contrast (reported by default):
```

```
run did continuous on cont2 using covariates using robust
  using clustered by "id" na omit

# Local ACR(d): focus on units with post dose near d = 3.0
run did continuous on cont2 at level 3.0 with bandwidth 0.5
  using robust na omit
```

Interpretation

- **Binary contrast** $\widehat{\beta}$: average shift in ΔY when moving from zero to positive post dose.
- **Global** $\widehat{\tau}$: average effect per unit increase in ΔD from pre to post.
- **Local** $\widehat{\tau}(d)$: marginal "nudge" effect around dose level d; informative when effects vary with the level of dose.

Notes

- Always pair with `check strong parallel trends`; if it fails, additional structure is needed for causal interpretation.
- The local ACR requires sufficient mass within $|D_{i1} - d| \leq h$; otherwise it is skipped with a warning.
- With multiple pre/post values configured, the routine uses the *last* pre and the *first* post period when forming differences.