Title

didregress — Difference-in-differences estimation

Description

didregress estimates the average treatment effect on the treated (ATET) from observational data by difference-in-differences (DID) or difference-in-difference-in-differences (DDD). The ATET of a binary or continuous treatment on a continuous outcome is estimated by fitting a linear model with time and group fixed effects. The DID and DDD estimation performed by didregress can be applied to data comprising repeated cross-sections in which different groups of individuals are observed at each time period.

xtdidregress estimates the ATET from observational data by DID or DDD for panel data. The ATET of a binary or continuous treatment on a continuous outcome is estimated by fitting a linear model with time and panel fixed effects.

Quick start

DID estimate of the ATET of treat1 on outcome y1 modeled using covariates x1 and x2, and grpvar1 and tvar fixed effects, with the treatment occurring at the grpvar1 and tvar levels

    didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar)

As above, but compute wild cluster–bootstrap p-values and confidence intervals with grpvar1 as the clustering variable

    didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
        wildbootstrap

Aggregate data at the grpvar1 and tvar levels to estimate the ATET

    didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
        aggregate(standard)

As above, but use the Donald and Lang (2007) method to compute the ATET and standard errors

    didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
        aggregate(dlang)

DDD estimate of the ATET of treat2 on outcome y2 modeled using covariates x1 and x2 and fixed effects defined by two-way interactions for grpvar1, grpvar2, and tvar, with the treatment occurring at the grpvar1, grpvar2, and tvar levels

    didregress (y2 x1 x2) (treat2), group(grpvar1 grpvar2) time(tvar)

DID estimate of ATET of treat3 on outcome y3 using xtset data; y3 modeled using covariates x1 and x2, and individual (panel) and tvar fixed effects, with the treatment occurring at the grpvar1 and tvar levels

    xtdidregress (y3 x1 x2) (treat3), group(grpvar1) time(tvar)

Menu

didregress

Statistics > Treatment effects > Continuous outcomes > Difference in differences

xtdidregress

Statistics > Treatment effects > Continuous outcomes > Difference in differences (FE)

Statistics > Longitudinal/panel data > Difference in differences

Syntax

DID with repeated cross-sectional data

> didregress (*ovar omvarlist*) (*tvar* [, continuous]) [*if*] [*in*] [*weight*],
>
> > group(*groupvars*) [time(*timevar*) *options*]

DID with longitudinal data

> xtdidregress (*ovar omvarlist*) (*tvar* [, continuous]) [*if*] [*in*] [*weight*],
>
> > group(*groupvars*) [time(*timevar*) *options*]

ovar is the outcome of interest.

omvarlist specifies the covariates in the outcome model and may contain factor variables; see [U] **11.4.3 Factor variables**.

tvar must be a binary variable indicating observations subject to treatment or a continuous variable measuring treatment intensity.

groupvars are categorical variables that indicate the group level at which the treatment occurs. At least one group variable must be specified. If *timevar* is specified, at most two group variables may be specified. If *timevar* is not specified, at most three group variables may be specified.

timevar is a time variable. It must be specified if *groupvar* has only one variable.

options	Description
Model	
* **group**(*groupvars*)	specify group variables
‡ **time**(*timevar*)	specify time variable
nointeract	exclude group() and time() interactions
nogteffects	do not include group and time effects in the model
aggregate(*aggmethod*)	aggregate to the levels defined by interacting *groupvars* and *timevar*
wildbootstrap[(*wildopts*)]	compute confidence intervals and *p*-values with the wild bootstrap
SE/Robust	
vce(*vcetype*)	*vcetype* may be **cluster** *clustvar*, **robust**, **hc2**, or **bootstrap**
Reporting	
level(*#*)	set confidence level; default is level(95)
aequations	display auxiliary-equation results
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
coeflegend	display legend instead of statistics

*group(*groupvars*) is required.

‡ time(*timevar*) is required when only one group is specified.

by, collect, and statsby are allowed; see [U] **11.1.10 Prefix commands**.

fweights, aweights, and pweights are allowed; see [U] **11.1.6 weight**.

coeflegend does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

aggmethod	Description
standard	aggregate data and fit model
dlang[, *dlopt*]	aggregate data and fit model using Donald and Lang method

wildopts	Description
errorweight(*edtype*)	specify the error weight type *edtype*; default is errorweight(rademacher)
reps(*#*)	perform *#* wild bootstrap replications; default is reps(1000)
rseed(*#*)	set random-number seed to *#*
blocksize(*#*)	perform wild bootstrap in blocks of *#* replications

Options

[Model]

group(*groupvars*) specifies group variables. It indicates the group level at which the treatment occurs. *groupvars* may be, for example, states, counties, or hospitals. *groupvars* define the group levels for which group effects are included in the model used to perform DID estimation and for which group interactions are included in the model used to perform DDD estimation. group() also defines the level clustering for the default cluster–robust standard errors. group() is required.

You may specify at most two group variables if you also specify a time variable in the `time()` option or at most three group variables if no time variable is specified.

`time(`*timevar*`)` specifies the time variable. You may specify `time()` when one or two group variables are specified in the `group()` option. `time()` is required when only one group variable is specified in the `group()` option.

`nointeract` excludes *groupvars* and *timevar* interactions from being included in the model. By default, `didregress` and `xtdidregress` include group and time interactions for your specification if there is more than one group variable.

`nogteffects` specifies to not include group and time effects. By default, `didregress` adds group and time dummies to the regression specification. By default, `xtdidregress` adds time dummies to the fixed-effects specification.

`aggregate(`*aggmethod*`)` fits the model by aggregating data at the *groupvars* and *timevar* levels. *aggmethod* may be either `standard` or `dlang[ , `*dlopt*`]`.

 `standard` specifies that aggregation is performed using the standard aggregation method. In this case, a regression model is fit of the original outcome on covariates that vary within levels of the *groupvars* and *timevar* interaction. The estimates of the group-time level effects from this regression are used to construct a new dependent variable. These effects along with the remaining covariates are then aggregated to the level of the of the *groupvars* and *timevar* interaction. The final results are obtained by regressing the estimated group-time level effects on the remaining covariates with this aggregated dataset and estimating group-level cluster–robust standard errors.

 `dlang[ , `*dlopt*`]` aggregates data using methods proposed by Donald and Lang (2007). *dlopt* may be either `constant` or `varying`.

  `constant` requests that standard errors be estimated using the standard ordinary least-squares method, as suggested by Donald and Lang (2007). With this method, as with the `standard` aggregation method, a single regression model is fit in the first step, so the coefficient estimates are constant across levels of the *groupvars* and *timevar* interaction.

  `varying` specifies that the aggregation method allow for varying coefficients on the covariates. Specifically, in this case, separate regression models of the original outcome on covariates that vary within levels of *groupvars* and *timevar* are fit for each level of the *groupvars* and *timevar* interaction. Thus, this method allows the coefficients on these variables to vary. The constant from each of these regressions forms the new dependent variable. The final regression and standard error computations are equivalent to those used by the `constant` method.

`wildbootstrap[ (`*wildopts*`)]` computes confidence intervals and *p*-values with the wild bootstrap. The wild bootstrap is constructed imposing the null hypothesis that the ATET is 0; that is, it is a restricted wild bootstrap. Confidence intervals are computed separately from the *p*-values. The bounds of the confidence interval are computed using a bisection optimization algorithm described in *Methods and formulas*. *wildopts* are `errorweight(`*edtype*`)`, `reps(#)`, `rseed(#)`, and `blocksize(#)`.

 `errorweight(`*edtype*`)` defines the error weight used to draw residuals from the wild bootstrap. *edtype* is one of `rademacher` (the default), `mammen`, `webb`, `normal`, or `gamma`.

  `rademacher` multiplies the residuals at each bootstrap replication with a randomly generated variable that takes the value of 1 with probability 0.5 and the value of -1 with probability 0.5. `errorweight(rademacher)` is the default.

  `mammen` multiplies the residuals at each bootstrap replication with a randomly generated variable that takes the value of $1-\phi$ with probability $\phi/\sqrt{5}$ and ϕ otherwise, where $\phi = (1+\sqrt{5})/2$.

webb multiplies the residuals at each bootstrap replication with a randomly generated variable that takes the values $-\sqrt{3/2}$, $-\sqrt{2/2}$, $-\sqrt{1/2}$, $\sqrt{1/2}$, $\sqrt{2/2}$, and $\sqrt{3/2}$, each with probability $1/6$.

normal multiplies the residuals at each bootstrap replication with a randomly generated normal distribution variable with the first four moments given by 0, 1, 0, and 3.

gamma multiplies the residuals at each bootstrap replication with a randomly generated gamma distribution variable with shape parameter 4 and scale parameter $1/2$.

reps(#) performs # wild bootstrap replications. The default is reps(1000).

rseed(#) sets the random-number seed to #.

blocksize(#) specifies that the wild bootstrap be performed in blocks, with # replications per block. The wild bootstrap computation requires a matrix with dimensions (# groups) × (# replications). If this is too large, you can reduce the matrix to (# groups) × (# block size) and loop (# replications)/(# block size) times. When the same random seed is set, using a different block size does not change the numerical results; it only modifies the computation method. The block size must be less than or equal to the number of bootstrap replications.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that allow for intragroup correlation (cluster *clustvar*), that are robust to some kinds of misspecification (robust), that are bias-corrected cluster robust using the degrees-of-freedom adjustment proposed by Bell and McCaffrey (2002) (hc2), and that use bootstrap sampling done at the group level (bootstrap); see [R] *vce_option*.

vce(cluster *clustvar*), the default, uses the first variable specified in the group(*groupvars*) option.

vce(hc2) specifies bias-corrected cluster–robust standard errors with the degrees-of-freedom adjustment proposed by Bell and McCaffrey (2002). As with vce(hc2) in [R] **regress**, the residuals are rescaled by the projection matrix to improve the small-sample properties of the variance estimates. For more details, see *Methods and formulas*.

Specifying vce(robust) is equivalent to specifying vce(cluster *clustvar*), where *clustvar* is the first variable specified in the group(*groupvars*) option.

Reporting

level(#); see [R] **Estimation options**.

aequations specifies that the results for the outcome-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

The following option is available with didregress and xtdidregress but is not shown in the dialog box:

coeflegend; see [R] **Estimation options**.

Remarks and examples   stata.com

Remarks are presented under the following headings:

Introduction

DID is one of the most venerable causal inference methods used by researchers. DID estimates the average treatment effect on the treated group (ATET).

To obtain the ATET using DID, one must compute the difference of the mean outcome for the treatment and the control groups before and after the treatment. This eliminates time-invariant unobservable group characteristics that confound the ATET; however, this is not enough to identify an effect. There may be time-varying unobservable confounders with an effect on the treatment group even after we control for time-invariant unobservable group characteristics. DID eliminates time-varying confounders by including a control group that is subject to the same time-varying confounders as the treatment group.

The ATET is then consistently estimated, differencing the mean outcome for the treatment and control groups over time to eliminate time-invariant unobservable characteristics and also differencing the mean outcome of these groups to eliminate time-varying unobservable effects common to both groups. These two differences give the DID method its name and highlight its intuitive appeal. More appealing is the fact that you can get the effect of interest, the ATET, from one parameter in a linear regression.

Below, we illustrate how to use didregress and xtdidregress. For more information about the methods used below, see [TE] **DID intro**. For general discussions of the DID methodology, see Angrist and Pischke (2009, 2015), Blundell and Dias (2009), Imbens and Wooldridge (2009), Lechner (2011), Abadie and Cattaneo (2018), and Wing, Simon, and Bello-Gomez (2018) and the references therein.

DID estimation

▷ Example 1: Fitting a DID model

A health provider is interested in studying the effect of a new hospital admissions procedure on the satisfaction of patients. The provider has monthly data on patients from January to July. The new admissions procedure was implemented in April by hospitals that were under new management. Of the 46 hospitals in the study, 18 implemented the new procedure.

The health provider will use a DID regression to analyze the effect of the new admissions procedure on the hospitals that participated in the program. The outcome of interest is patient satisfaction, satis, which is recorded as an average of the responses to a set of four questions asked to patients. satis may take values between 0 and 10, where 10 is the greatest possible level of satisfaction and 0 is utter disappointment. The procedure variable marks the treated observations; it is 1 if a surveyed individual was admitted to the hospital using the new procedure after March and 0 otherwise. To get the ATET on the outcome satis, we type

```
. didregress (satis) (procedure), group(hospital) time(month)
```

The first set of parentheses is used to specify the outcome of interest followed by the covariates in the model. In this case, there are no covariates, just the outcome, `satis`. The second set of parentheses is used to specify the binary variable that indicates the treated observations, `procedure`. The `group()` and `time()` options are used to construct group and time fixed effects that are included in the model. The variable specified in `group()` is also important because it defines the level of clustering for the default cluster–robust standard errors; in this case, we cluster at the hospital level. The results from this command are as follows:

```
. use https://www.stata-press.com/data/r17/hospdd
(Artificial hospital admission procedure data)
. didregress (satis) (procedure), group(hospital) time(month)
Number of groups and treatment time

Time variable: month
Control:       procedure = 0
Treatment:     procedure = 1
```

	Control	Treatment
Group		
hospital	28	18
Time		
Minimum	1	4
Maximum	1	4

```
Difference-in-differences regression              Number of obs = 7,368
Data type: Repeated cross-sectional

                            (Std. err. adjusted for 46 clusters in hospital)
```

satis	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]
ATET						
procedure (New vs Old)	.8479879	.0321121	26.41	0.000	.7833108	.912665

Note: ATET estimate adjusted for group effects and time effects.

The first table gives information about the treated and control groups and about treatment timing. The first segment with the title Group tells us the number of treated and control hospitals: 28 hospitals were using the old procedure and 18 were using the new one. The second segment of the table gives information about the first time we observe hospitals in the control group and the first time we observe the treatment (the new admission procedure) for hospitals in the treatment group. In this example, all hospitals that adopted the new procedure did so in April, time period 4. If some hospitals had adopted the policy later, the minimum and maximum time of first treatment would differ.

The ATET is 0.85, almost a 1-point increase in satisfaction relative to the case where none of the treated hospitals enacted the new procedure. In other words, if the hospitals that implemented the new admission procedure had not done so, their satisfaction ratings would be lower by almost one point on average.

We now explore whether the trajectories of `satis` are parallel for the control and treatment groups prior to the date when the new procedure was implemented. We are checking what is known as the parallel-trends or common-trends assumption, an important assumption of the DID model, as discussed in [TE] **DID intro**. A visual diagnostic of this assumption can be obtained by plotting the means of

the outcome over time for both groups or by visualizing the results of the linear-trends model. We can perform both of these diagnostic checks by using `estat trendplots`. To obtain figure 1 below, we type
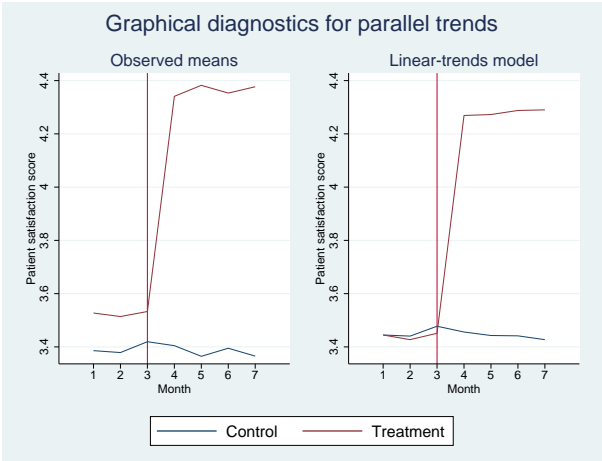
```
. estat trendplots
```


Figure 1

The graph seems to indicate that the parallel-trends assumption is satisfied. Prior to the policy implementation, treated and control hospitals followed a parallel path.

We could also perform a test to see if the trajectories are parallel by augmenting our original model to include variables representing time trends before and after the treatment for both groups of hospitals. The linear-trends model estimates a coefficient for the differences in linear trends prior to treatment, and if that coefficient is 0, the linear pretreatment trends are parallel. Otherwise, identification of the ATET may become questionable.

We can perform this test by using `estat ptrends`:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 45) =    0.55
Prob > F = 0.4615
```

We do not have evidence to reject the null hypothesis of parallel trends in this case. Both the test and the graphical analysis support the parallel-trends assumption and, therefore, our ATET estimate.

◁

▷ Example 2: Fitting a DDD model

The results in example 1 could come into question if they could be the consequence of other unobserved variables rather than the consequence of the new hospital admissions procedure. The health provider administrators believe that responses to the survey are related to the frequency of individuals' hospital visits. The patients may have unobserved characteristics that affect both how frequently they visit the hospital and how they feel about the admissions procedure. In other words, there might be unobserved characteristics that confound the effect of the new hospital admissions procedure. The administrators decide to obtain the ATET by using a DDD model. They want to estimate the average treatment effect on patients who visit the hospital with high or very high frequencies.

To do this, we will first create a new variable `hightrt` to be our new treatment identifier. Observations are now marked as treated (`hightrt = 1`) if hospital visit frequency by an individual is high or very high (`frequency = 3` or `4`) and if the hospital implemented the new admissions procedure in April.

```
. generate hightrt = procedure==1 & (frequency==3 | frequency==4)
. label define trt 0 "Untreated" 1 "Treated"
. label values hightrt trt
```

The DDD model will incorporate both hospital and frequency of usage effects as well as their interaction with time effects. To fit the model, we incorporate a new group variable, `frequency`:

```
. didregress (satis) (hightrt), group(hospital frequency) time(month)
  (output omitted )
```

Number of groups and treatment time

Time variable: month
Control: hightrt = 0
Treatment: hightrt = 1

	Control	Treatment
Group		
hospital	28	18
frequency	2	2
Time		
Minimum	1	4
Maximum	1	4

Triple-differences regression   Number of obs = 7,368
Data type: Repeated cross-sectional

(Std. err. adjusted for 46 clusters in hospital)

satis	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]
ATET						
hightrt (Treated vs Untreated)	.764154	.0402603	18.98	0.000	.6830655	.8452425

Note: ATET estimate adjusted for group effects, time effects, and group- and
      time-effects interactions.

The omitted output after the command corresponds to the factor-variable interactions that include the base categories. This is common when you fit DDD models that by default include group interactions and group and time interactions.

The first table above has information on the second group variable, `frequency`, for which low and medium frequencies are controls and high and very high frequencies are treated. The second table shows that the ATET is now smaller, but the policies still increase satisfaction.

◁

▷ Example 3: DID for panel-data model

Moser and Voena (2012) look at the effect of compulsory licensing on domestic inventions. Compulsory licensing allows firms in developing countries to produce foreign inventions without the consent of foreign patent owners. Having access to foreign technology may discourage domestic inventions, but it could also enhance local production.

Moser and Voena consider legislation that occurred during World War I called Trading With the Enemy Act (TWEA). By 1919, German-owned patents were systematically licensed to U.S. companies. In Moser and Voena (2012), the treated observations correspond to a subclass in the chemical industry that was granted at least one of the TWEA patents after 1918, as reported by United States Patent and Trademark Office (USPTO). A subclass is a group of firms in an industry that employ similar technologies, as defined by the USPTO.

The outcome of interest is the number of patents granted to inventors from the U.S. in that subclass, uspatents. This measures domestic innovation. Moser and Voena also include the number of non-TWEA patents granted to the subclass that were from foreign inventors, fpatents. fpatents measure innovation in the subclass that is not from U.S. inventors. We observe the same subclasses at each point in time from 1875 to 1939. Thus, we have a panel dataset.

To fit the model, we first xtset our data at the subclass level, classid:

```
. use https://www.stata-press.com/data/r17/patents
(Excerpt from Moser and Voena (2012))
. xtset classid
Panel variable: classid (balanced)
```

Below, we fit a DID model for the number of patents granted to U.S. inventors in a subclass, controlling for the number of non-TWEA patents granted to foreign inventors. The treatment indicator gotpatent is 1 if the subclass received a TWEA patent after 1918 and is 0 otherwise. The model includes year fixed effects and subclass fixed effects.

```
. xtdidregress (uspatents fpatents) (gotpatent), group(classid) time(year)

Number of groups and treatment time

Time variable: year
Control:        gotpatent = 0
Treatment:      gotpatent = 1
```

	Control	Treatment
Group		
classid	6912	336
Time		
Minimum	1875	1919
Maximum	1875	1919

```
Difference-in-differences regression                   Number of obs = 471,120
Data type: Longitudinal
```

(Std. err. adjusted for 7,248 clusters in classid)

uspatents	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]
ATET					
gotpatent (Patent vs None)	.150516	.0356081	4.23	0.000	.0807137 .2203183

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

The ATET is 0.15, which means that in subclasses that were awarded one or more patents, domestic inventors produced an average of 0.15 additional patents after the TWEA compared with the scenario in which no patents are awarded for those subclasses.

◁

Graphical diagnostics and tests

▷ Example 4: DID diagnostic graphs and tests

As illustrated in example 1, when conducting a DID study, it is common to complement the regression analysis with graphical diagnostics and tests that provide evidence of whether an estimated effect can be given a causal interpretation. As discussed in [TE] DID intro, we would like to observe that the treated and control groups had mean outcomes that evolved similarly to each other over time prior to the treatment. This is usually referred to as a parallel-trends or common-trends assumption. We would also like to ascertain that neither the control nor the treatment group changed their behavior in anticipation of the treatment. This is assessed using a Granger-type test.

Below, we use simulated data to illustrate the diagnostics and tests available after didregress and xtdidregress. Simulated data helps us know exactly what we should expect and how to interpret it.

Suppose we have a panel dataset with 10 time points, denoted by t1, where a treatment has taken place between $t1 = 5$ and $t1 = 6$. We have a set of covariates, x1 and x2, and an outcome, y1. The outcome could be something like patient satisfaction, as in example 1, or the number of patents filed by U.S. inventors, as in example 3.

We fit the model:

```
. use https://www.stata-press.com/data/r17/parallelt
(Simulated data to test parallel-trends assumption)
. xtset id1
Panel variable: id1 (unbalanced)
. xtdidregress (y1 c.x1##c.x2) (treated1), group(id1) time(t1)
Number of groups and treatment time
Time variable: t1
Control:       treated1 = 0
Treatment:     treated1 = 1
```

	Control	Treatment
Group		
id1	102	98
Time		
Minimum	1	6
Maximum	1	6

```
Difference-in-differences regression            Number of obs = 2,000
Data type: Longitudinal
                            (Std. err. adjusted for 200 clusters in id1)
```

y1	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]
ATET						
treated1						
(Treated						
vs						
Untreated)	.5069426	.0220218	23.02	0.000	.4635166	.5503686

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

Is this result valid? We can first explore the assumption of parallel trends graphically, comparing the trajectories of the outcome variable for the control and treatment groups prior to the date of treatment. We can check this assumption by plotting the means of the outcome over time for both groups or by visualizing the results of the linear-trends model. We can perform both of these diagnostic checks by using estat trendplots.
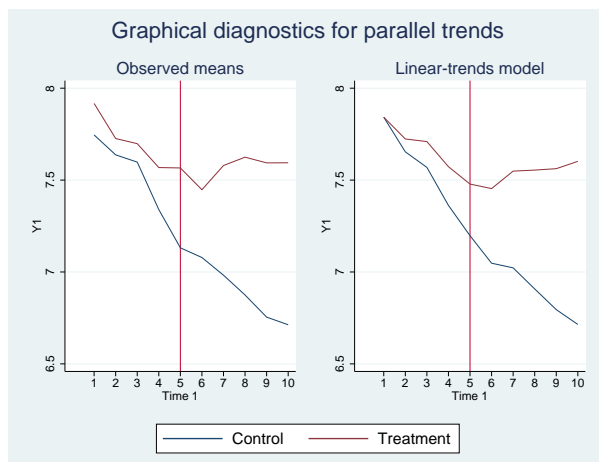
```
. estat trendplots
```


Figure 2

Looking at the plotted observed means (left side of figure 2), the outcome trajectories in the control and treatment groups prior to the treatment are somewhat different. While we can observe declining trends in both groups, it looks as though the decline is progressing more rapidly in the control group, especially between time points three and five. We can get a clearer picture of this by looking at the results of the linear-trends model on the right side of figure 2. The group-level trajectories are shown with respect to a common reference point, t1 = 1, which makes it easy to discern whether they are parallel. In this case, we can see that they are not. The differences between the treatment and control groups are growing larger over time up to t1 = 5 (posttreatment time periods are not relevant when assessing the parallel-trends assumption). Judging by figure 2 alone, we should be concerned about whether the parallel-trends assumption holds for our effect estimate.

A more formal way to assess whether the pretreatment trajectories are parallel is to perform a test on the linear-trends model coefficient that captures the differences in the trends between treated and controls. If the pretreatment trends are actually linear in both groups, then this coefficient will be 0 because there are no differences in the slopes between the two groups. Thus, by testing this coefficient against 0, we have a test of the null hypothesis that the pretreatment period trajectories are parallel. We can perform this test by using `estat ptrends`:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 199) =   39.97
 Prob > F = 0.0000
```

We reject the null hypothesis of the linear trends being parallel.

We can also think of nonparallel as an indication of an anticipatory treatment effect. We saw that the trends were not parallel before the treatment took place, which could indicate a treatment effect even before the treatment is implemented. Thus, another way to state our parallel-trends assumption is that there should be no treatment effect in anticipation of the treatment. To test this assumption, we could fit a Granger-type causality model where we augment our model with dummies that indicate future treatment status for each time period prior to the treatment. A joint test of the coefficients on these dummies against 0 can be used as a test of the null hypothesis that no anticipatory effects have taken place. We can perform this test by using `estat granger`:

```
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(4, 199) =   18.17
 Prob > F = 0.0000
```

The result suggests that we reject the null hypothesis of no anticipatory effects prior to treatment. Thus, based on the results of both `estat ptrends` and `estat granger`, we conclude that we should be concerned about identification of ATET.

Notice that the parallel-trends F test consumes only 1 numerator degree of freedom, while the Granger causality F test consumes 4. That is because we have five pretreatment time periods and thus four coefficients to test. In this example, both tests would be appropriate, but the parallel-trends test has higher statistical power. However, as we will see below, the less-powered Granger test is more flexible and can be used in situations where the differences between treatment and control groups are nonlinear in pretreatment periods.

◁

▷ Example 5: DID diagnostics and tests with nonlinear mean outcome differences

We now look at a different outcome and policy using the same simulated dataset as in example 4. We fit the same model but with the new variables:

```
. xtset id2
Panel variable: id2 (balanced)
. xtdidregress (y2 c.z1##c.z2) (treated2), group(id2) time(t2)
Number of groups and treatment time

Time variable: t2
Control:       treated2 = 0
Treatment:     treated2 = 1
```

	Control	Treatment
Group		
id2	480	520
Time		
Minimum	1	6
Maximum	1	6

```
Difference-in-differences regression                      Number of obs = 10,000
Data type: Longitudinal
                                    (Std. err. adjusted for 1,000 clusters in id2)
```

		Robust				
y2	Coefficient	std. err.	t	P>|t|	[95% conf. interval]	
ATET						
treated2						
(Treated						
vs						
Untreated)	.2636651	.0097188	27.13	0.000	.2445936	.2827367

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

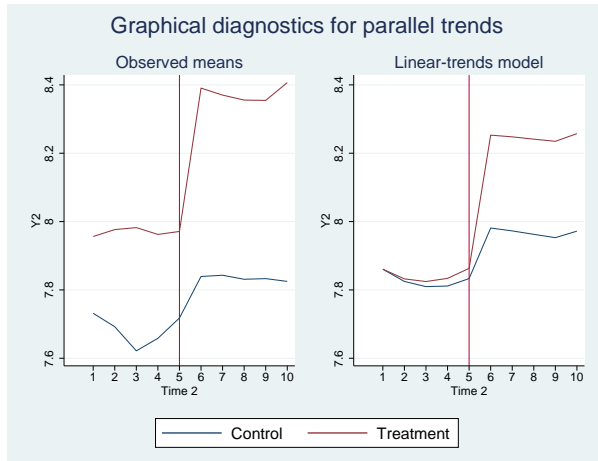Here is the diagnostic plot produced by `estat trendplots`:

Figure 3

Inspecting the observed means over the pretreatment time periods, we can see that no considerable change has occurred in the outcome of the treatment group over time. For the control group, on the other hand, we observe a somewhat U-shaped trajectory in advance of the treatment. However, if we look at the plot from the linear-trends model, both trajectories appear somewhat U-shaped, and the trends appear to be almost parallel. Indeed, if we use `estat ptrends`, we fail to reject the null hypothesis of parallel trends:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 999) =    2.13
 Prob > F = 0.1446
```

The Granger causality test can handle cases in which the trajectories are nonlinear and for which `estat ptrends` will fail to reject the null hypothesis when it should:

```
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(4, 999) =    9.86
 Prob > F = 0.0000
```

We correctly reject the null hypothesis of no effect in anticipation of the treatment.

◁

▷ Example 6: Diagnostics and tests when parallel-trends assumption is satisfied

Finally, we look at a case where the assumption of parallel trends is satisfied. Again, we use the simulated dataset from example 4. We fit the following model:

```
. xtset id3
Panel variable: id3 (balanced)
. xtdidregress (y3 c.w1##c.w2) (treated3), group(id3) time(t3)
Number of groups and treatment time
Time variable: t3
Control:        treated3 = 0
Treatment:      treated3 = 1
```

	Control	Treatment
Group		
id3	502	498
Time		
Minimum	1	6
Maximum	1	6

```
Difference-in-differences regression                 Number of obs = 10,000
Data type: Longitudinal
                               (Std. err. adjusted for 1,000 clusters in id3)
```

	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]
ATET					
treated3 (Treated vs Untreated)	.4996049	.0102458	48.76	0.000	.4794991 .5197107

Note: ATET estimate adjusted for covariates, panel effects, and time effects.
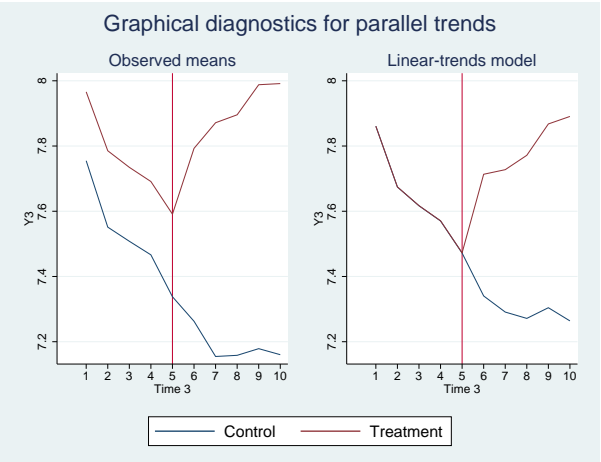
Using `estat trendplots`, we obtain

Figure 4

This time, the trajectories of the observed means appear to be parallel before the treatment occurs. Both follow a declining trend up to the last pretreatment time point. Looking at the results from the linear-trends model, the pretreatment trajectories appear to be the same. Using `estat ptrends` yields the following result:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 999) =    0.00
 Prob > F = 0.9688
```

We cannot reject the null hypothesis of parallel linear trends. Likewise, using `estat granger`, we do not reject the null hypothesis of the absence of anticipatory effects:

```
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(4, 999) =    0.52
 Prob > F = 0.7220
```

◁

▷ Example 7: Time-specific treatment effects

As we have seen in the previous examples, we can use `estat granger` to test for treatment effects occurring prior to the point at which treatment has been administered. Beyond the particular purpose of this test, however, it can be instructive to inspect not only pretreatment effects but also posttreatment effects that are allowed to vary over time. That is, rather than relying on a single treatment-effect estimate that is assumed to be constant over time, it can be instructive to check whether treatment effects change as time unfolds. We can fit a model that includes lags and leads of an indicator for the time period in which the treatment began. The coefficients on the lags can be used to assess whether there is any change in the treatment effect during posttreatment time periods. We can use the postestimation command `estat grangerplot` to fit such a model and plot the resulting coefficients. This model is also sometimes referred to as an event study model in the literature; see Schmidheiny and Siegloch (2019) and Clarke and Tapia-Schythe (2021).

We start by fitting the following DID model:

```
. use https://www.stata-press.com/data/r17/hospdd
(Artificial hospital admission procedure data)
. didregress (satis) (procedure), group(hospital) time(month)
```

Number of groups and treatment time

```
Time variable: month
Control:       procedure = 0
Treatment:     procedure = 1
```

	Control	Treatment
Group		
hospital	28	18
Time		
Minimum	1	4
Maximum	1	4

Difference-in-differences regression   Number of obs = 7,368
Data type: Repeated cross-sectional

(Std. err. adjusted for 46 clusters in hospital)

satis	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]
ATET						
procedure (New vs Old)	.8479879	.0321121	26.41	0.000	.7833108	.912665

Note: ATET estimate adjusted for group effects and time effects.

We can now simply use `estat grangerplot` to plot pre- and postintervention treatment effects:

```
. estat grangerplot
```

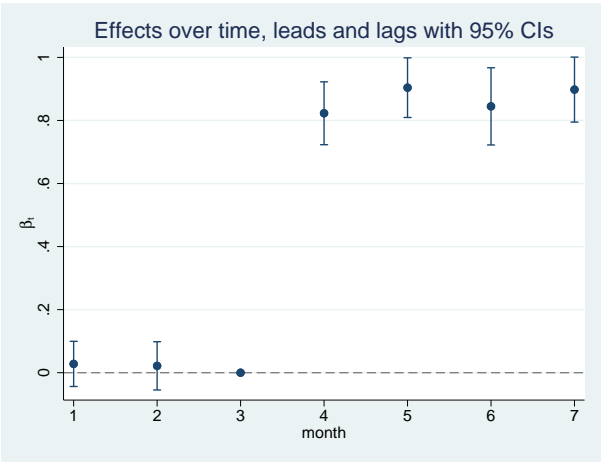This yields the following graph:

Figure 5

By default, `estat grangerplot` uses all available leads and lags, and in this case, we have estimates for two leads and four lags. For identification purposes, we must omit one of the indicator variables. By default, `estat grangerplot` omits the first lead, which sets the coefficient to 0 that corresponds to the time period prior to treatment administration. Thus, the plotted effects are normalized with respect to that time period. If we wanted, we could change the baseline period by using the `baseline()` option. We can see that the coefficients on the leads that correspond to the first two months are both close to 0, which shows that there are no substantial effects in the pretreatment era. The posttreatment effects for months 4 to 7 range between around 0.8 and 0.9. When we account for the uncertainty of these estimates, as indicated by the plotted 95% confidence intervals, it appears as though the treatment effects are rather stable over time in this example.

Rather than just plotting the coefficients, we could be interested in looking at a table with the numeric results. We can obtain the underlying results by specifying the `verbose` option. We are also using the `nodraw` option to see only the numeric results and not draw the graph again:

```
. estat grangerplot, nodraw verbose
```

Linear regression, absorbing indicators   Number of obs   =   7,368
Absorbed variable: hospital   No. of categories =   46
                                                   F(12, 45)   =   94.68
                                                   Prob > F   = 0.0000
                                                   R-squared   = 0.5334
                                                   Adj R-squared   = 0.5298
                                                   Root MSE   = 0.7240

(Std. err. adjusted for 46 clusters in hospital)

satis	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]	
month						
February	-.007044	.0263953	-0.27	0.791	-.0602068	.0461188
March	.0335696	.0255925	1.31	0.196	-.0179764	.0851156
April	.0187852	.0250623	0.75	0.457	-.0316927	.0692632
May	-.0211152	.0269569	-0.78	0.438	-.0754092	.0331788
June	.0091208	.0179016	0.51	0.613	-.026935	.0451766
July	-.0203444	.0306266	-0.66	0.510	-.0820296	.0413407
_lead3	.027897	.035569	0.78	0.437	-.0437426	.0995367
_lead2	.0217322	.0380076	0.57	0.570	-.054819	.0982833
_lag0	.8228153	.0494933	16.62	0.000	.7231307	.9224999
_lag1	.9040498	.0469682	19.25	0.000	.8094511	.9986486
_lag2	.844724	.0608006	13.89	0.000	.7222654	.9671826
_lag3	.8978885	.0511588	17.55	0.000	.7948494	1.000928
_cons	3.433074	.0198449	173.00	0.000	3.393104	3.473044

The coefficients on the variables that begin with _lead and _lag are the ones that are plotted in our graph.

As an interesting side note, notice that our original DID model is a special case of the model that we fit with `estat grangerplot`. Specifically, it is the model that includes only a single lead and no lags. To reproduce our original DID result, we can use the `nleads()` and `nlags()` options:

```
. estat grangerplot, nodraw verbose nleads(1) nlags(0)
```

```
Linear regression, absorbing indicators          Number of obs    =  7,368
Absorbed variable: hospital                      No. of categories =     46
                                                 F(7, 45)          = 138.73
                                                 Prob > F          = 0.0000
                                                 R-squared         = 0.5333
                                                 Adj R-squared     = 0.5299
                                                 Root MSE          = 0.7238
```

(Std. err. adjusted for 46 clusters in hospital)

satis	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]	
month						
February	-.0096077	.0184317	-0.52	0.605	-.0467311	.0275158
March	.0219686	.018251	1.20	0.235	-.0147907	.0587279
April	-.0032839	.0221028	-0.15	0.883	-.0478013	.0412335
May	-.0094027	.0232399	-0.40	0.688	-.0562103	.0374048
June	-.0038375	.0190634	-0.20	0.841	-.0422332	.0345581
July	-.0111941	.0230029	-0.49	0.629	-.0575244	.0351361
_lag0	.8479879	.0321121	26.41	0.000	.7833108	.912665
_cons	3.444675	.011354	303.39	0.000	3.421807	3.467543

As we can see, the results match the ones we obtained earlier with didregress.

Notice also that the model used by estat granger is a special case as well; it is the model with all available leads and no lags. To re-create the test result from estat granger, we use the post option to store the results:

```
. estat grangerplot, nodraw verbose post nlags(0)
```

```
Linear regression, absorbing indicators          Number of obs    =  7,368
Absorbed variable: hospital                      No. of categories =     46
                                                 F(9, 45)          = 113.78
                                                 Prob > F          = 0.0000
                                                 R-squared         = 0.5333
                                                 Adj R-squared     = 0.5298
                                                 Root MSE          = 0.7239
```

(Std. err. adjusted for 46 clusters in hospital)

satis	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]	
month						
February	-.007044	.0263899	-0.27	0.791	-.0601959	.0461079
March	.0335696	.0255873	1.31	0.196	-.0179658	.085105
April	.0002573	.0243547	0.01	0.992	-.0487955	.0493101
May	-.0058616	.0249982	-0.23	0.816	-.0562105	.0444874
June	-.0002964	.0199414	-0.01	0.988	-.0404605	.0398678
July	-.007653	.0251828	-0.30	0.763	-.0583738	.0430679
_lead3	.027897	.0355617	0.78	0.437	-.043728	.099522
_lead2	.0217322	.0379998	0.57	0.570	-.0548033	.0982676
_lag0	.8673694	.0424929	20.41	0.000	.7817844	.9529544
_cons	3.433074	.0198408	173.03	0.000	3.393113	3.473035

We can now test whether the coefficients on the leads are jointly 0:

```
. test _lead3 _lead2
 ( 1)  _lead3 = 0
 ( 2)  _lead2 = 0
       F(  2,    45) =     0.33
            Prob > F =    0.7239
```

The test result indicates that we cannot reject the null hypothesis that the coefficients are jointly 0. We fit our original model again to compare the results with the ones from `estat granger`:

```
. quietly didregress (satis) (procedure), group(hospital) time(month)
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(2, 45) =   0.33
Prob > F = 0.7239
```

As we can see, the results are the same. Using `estat grangerplot` followed by `test` allows us to perform additional tests of lags and leads that are not available through `estat granger`.

◁

Specifying a 2-by-2 DID

▷ Example 8: Specifying a 2-by-2 DID

`didregress` and `xtdidregress` by default fit generalized DID models, also known as two-way fixed-effects models. Yet DID models sometimes are viewed from a two-period and two-group perspective, a 2-by-2 DID. You can also fit a 2-by-2 DID using `didregress` and `xtdidregress`. In fact, you will get equivalent results using a 2-by-2 DID or a generalized DID. The generalized DID is the default because it allows for a wider range of specifications that would not be feasible within the 2-by-2 framework, such as cases when the intervention occurs at different points in time.

In example 1, we got the ATET for hospitals that instituted a new admissions procedure. We typed the following:

```
. use https://www.stata-press.com/data/r17/hospdd
. didregress (satis) (procedure), group(hospital) time(month)
```

This implies that we are regressing `satis` on `procedure` and indicators for hospitals and for months. The indicators are created and added as regressors to our model by default. To fit a 2-by-2 model, we need to omit the hospital and month indicators and instead add an indicator for the period after treatment and an indicator for groups that are treated.

This is what we do below. We create the treatment-group indicator based on the fact that the variable `procedure` is 1 for individuals that experienced the new procedure and 0 otherwise; this variable identifies treated hospitals. We create the posttreatment indicator based on the fact that the new procedures are established after March, which has a value of 3 in the data.

```
. bysort hospital: egen treated = max(procedure)
. generate post = month>3
```

We now fit

```
. didregress (satis i.treated i.post) (procedure), nogteffects
> group(hospital) time(month)
```

Number of groups and treatment time

Time variable: month
Control: procedure = 0
Treatment: procedure = 1

	Control	Treatment
Group		
hospital	28	18
Time		
Minimum	1	4
Maximum	1	4

Difference-in-differences regression   Number of obs = 7,368
Data type: Repeated cross-sectional

                               (Std. err. adjusted for 46 clusters in hospital)

satis	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]
ATET					
procedure					
(New					
vs					
Old)	.8479879	.0320051	26.50	0.000	.7835263 .9124494

Note: ATET estimate adjusted for covariates.

We add the new indicators to our list of covariates and use the `nogteffects` option to exclude the group and time indicators that were included by default.

The point estimates are identical. What changes are the standard errors. They change because we have a different number of regressors.

◁

Standard-error considerations

▷ Example 9: Standard errors and data aggregation

In example 3, we had 336 treated subclasses and 7,248 subclasses in total. However, many studies face the challenge of having very few elements per group. In fact, sometimes the data analyzed consist of only two groups—the treatment group in which treatment is administered to members of the group and the controls group in which no treatment is administered. Think, for example, of an analysis at the state level where one state is treated and one state is controlled. As discussed in [TE] **DID intro**, these scenarios with few elements pose a challenge for inference. For a good discussion on these issues, see MacKinnon (2019).

`didregress` and `xtdidregress` provide alternatives in such cases. One alternative is to compute standard errors by using the wild cluster bootstrap. Another alternative is to use bias-corrected clustered standard errors with the degrees-of-freedom adjustment proposed by Bell and McCaffrey (2002). A final alternative is to aggregate your data and then compute effects and standard errors such as proposed by Donald and Lang (2007) or use bias-corrected standard errors.

Below, we explore these options using simulated data. We created a dataset with 2,000 individuals and five time periods. The treatment occurs at the county level and there are six counties. Two of the counties, county 1 and county 2, receive the treatment and the remaining four counties do not. The true value of the ATET is -1.0. As we see below, the groups are unbalanced.

```
. use https://www.stata-press.com/data/r17/smallg
(Simulated data with a small number of groups)

. tab county
```

County	Freq.	Percent	Cum.
1	715	7.15	7.15
2	2,570	25.70	32.85
3	3,410	34.10	66.95
4	2,285	22.85	89.80
5	920	9.20	99.00
6	100	1.00	100.00
Total	10,000	100.00	

First, we fit the model using the default standard errors, which perform well with many balanced groups but not necessarily well with data like these. We compare the other results with these results.

```
. didregress (outcome x i.b) (treated), group(county) time(year)
```

Number of groups and treatment time

```
Time variable: year
Control:       treated = 0
Treatment:     treated = 1
```

	Control	Treatment
Group		
county	4	2
Time		
Minimum	2011	2013
Maximum	2011	2013

```
Difference-in-differences regression                    Number of obs = 10,000
Data type: Repeated cross-sectional

                            (Std. err. adjusted for 6 clusters in county)
```

outcome	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]
ATET						
treated (Treated vs Untreated)	-.9394987	.0884134	-10.63	0.000	-1.166773 -.7122247

Note: ATET estimate adjusted for covariates, group effects, and time effects.

For this draw, the point estimates are reasonably close to the true value of -1.0, and the true values are contained inside the confidence interval.

Next, we use the wild cluster bootstrap. For a good introduction to the methodology, see Cameron, Gelbach, and Miller (2008), MacKinnon and Webb (2018), and Roodman et al. (2019). The wild cluster bootstrap works well in scenarios like the one above, where there are few groups.

We use the wild cluster bootstrap to construct p-values and confidence intervals, imposing the null hypothesis that the ATET is 0. We describe both computations in *Methods and formulas*. Here are the results:

```
. didregress (outcome x i.b) (treated), group(county) time(year)
> wildbootstrap(rseed(123) errorweight(webb))
computing 1000 replications

Confidence interval lower bound
........................
Confidence interval upper bound
......

Number of groups and treatment time

Time variable: year
Control:       treated = 0
Treatment:     treated = 1
```

	Control	Treatment
Group		
county	4	2
Time		
Minimum	2011	2013
Maximum	2011	2013

```
DID with wild-cluster bootstrap inference        Number of obs   = 10,000
                                                 No. of clusters =      6
                                                 Replications    =  1,000
Data type:    Repeated cross-sectional
Error weight: webb
```

outcome	Coefficient	t	P>|t|	[95.10% conf. interval]	
ATET					
treated					
(Treated vs Untreated)	-.9394987	-10.63	0.020	-1.248532	-.5621484

Note: 95.10% confidence interval is wider than requested.
Note: ATET estimate adjusted for covariates, group effects, and time effects.

Above, we first see the iterations used to find the confidence interval lower bound and upper bound. The optimization algorithm sometimes converges to a level that is below the test size. In such cases, the confidence interval is conservative. In the example above, instead of a 95% confidence interval, you obtain a 95.1% confidence interval. If there are t statistics that have exactly the same value across bootstrap replications, the algorithm will not solve exactly for the requested confidence level. We also use `errorweight()` with error weight `webb` to compute the wild bootstrap. This is best in cases with less than 10 groups, as suggested by Roodman et al. (2019). The results from the wild bootstrap suggest more uncertainty than the default confidence interval.

Now we look at two data aggregation methods. In both cases, data aggregation occurs in two steps. In the first step, we regress the outcome on the set of covariates that vary at the individual, group, and time levels. We then estimate the group–time fixed effects from this procedure to use in a second stage as the new dependent variable. In the second stage, we aggregate the remaining covariates and the new dependent variable at the group and time levels and run a regression. This aggregation strategy is described in more detail in Bertrand, Duflo, and Mullainathan (2004), Donald and Lang (2007), and Cameron and Miller (2015). We also describe the aggregation methods in *Methods and formulas*.

Below, we aggregate the data as described above and estimate bias-corrected standard errors by using the degrees-of-freedom adjustment suggested by Bell and McCaffrey (2002). This method of getting standard errors is computationally intensive, so it is well suited for cases where the dimensionality of the problem is reduced via aggregation.

```
. didregress (outcome x i.b) (treated), group(county) time(year)
> aggregate(standard) vce(hc2)
Computing degrees-of-freedom:
treated .
Number of groups and treatment time
Time variable: year
Control:      treated = 0
Treatment:    treated = 1
```

	Control	Treatment
Group		
county	4	2
Time		
Minimum	2011	2013
Maximum	2011	2013

```
Difference-in-differences regression              Number of obs   = 30
                                                  No. of clusters =  6
Data type:    Repeated cross-sectional
Aggregation:  Standard
```

outcome	Coefficient	Robust HC2 std. err.	t	P>|t|	[95% conf. interval]
ATET					
treated					
(Treated					
vs					
Untreated)	-.9958521	.1373277	-7.25	0.017	-1.566242 -.4254624

Note: ATET estimate adjusted for covariates, group effects, and time effects.

We see that the confidence intervals are again wider than with the default standard errors. It is also worth noticing that we went from 10,000 observations to 30. This is because we aggregated at the county and year levels. We have six counties and five years.

We could also aggregate our data and compute the t statistics using the degrees of freedom proposed by Donald and Lang (2007). This gives us

```
. didregress (outcome x i.b) (treated), group(county) time(year)
> aggregate(dlang)

Number of groups and treatment time

Time variable: year
Control:       treated = 0
Treatment:     treated = 1
```

	Control	Treatment
Group		
county	4	2
Time		
Minimum	2011	2013
Maximum	2011	2013

```
Difference-in-differences regression                 Number of obs = 30
Data type:    Repeated cross-sectional
Aggregation: Donald-Lang
```

outcome	Coefficient	Std. err.	t	P>|t|	[95% conf. interval]
ATET					
treated					
(Treated					
vs					
Untreated)	-.9958521	.1224496	-8.13	0.000	-1.248576 -.7431287

Note: ATET estimate adjusted for covariates, group effects, and time effects.

Again, we see wider confidence intervals.

Whenever you have few elements for each group and the groups are unbalanced, as in this example, you should be careful not to base your conclusions solely on the default cluster–robust standard errors. As was mentioned in [TE] **DID intro**, you should validate your conclusions by using one of the standard error computations suggested above.

◁

Default cluster–robust standard errors

▷ Example 10: Default cluster–robust standard errors with didregress and xtdidregress

We have stated that `didregress` was designed to handle repeated cross-sections and `xtdidregress` was designed to handle panel datasets. However, you can use `xtdidregress` when you have repeated cross-sections. For instance, for the hospital dataset in example 1, you may have typed

```
. use https://www.stata-press.com/data/r17/hospdd
(Artificial hospital admission procedure data)
. xtset hospital
Panel variable: hospital (unbalanced)
. xtdidregress (satis) (procedure), group(hospital) time(month)
Number of groups and treatment time
Time variable: month
Control:       procedure = 0
Treatment:     procedure = 1
```

	Control	Treatment
Group		
hospital	28	18
Time		
Minimum	1	4
Maximum	1	4

```
Difference-in-differences regression              Number of obs = 7,368
Data type: Longitudinal
                                (Std. err. adjusted for 46 clusters in hospital)
```

satis	Coefficient	Robust std. err.	t	P>|t|	[95% conf. interval]
ATET					
procedure					
(New					
vs					
Old)	.8479879	.0320138	26.49	0.000	.7835088 .9124669

Note: ATET estimate adjusted for panel effects and time effects.

You will get the same point estimate as you would get with `didregress` but different standard errors. The reason is that `xtdidregress` with cluster–robust standard errors does not count the group variables as regressors in the degrees-of-freedom correction used to compute standard errors. It relies on the asymptotic theory of fixed-effects regression, where the number of group effects are expected to grow with the sample size. Put differently, `xtdidregress` is using `xtreg, fe` to compute the default cluster–robust standard errors. `didregress` uses `areg`, which assumes the number of groups is fixed and counts them in the degrees-of-freedom computation.

As a practical matter, these standard errors are going to be close to each other when the number of observations per cluster is large, as is the case for this example. As the number of observations per cluster becomes smaller, however, the standard errors will differ more, with standard errors of `didregress` tending to be larger.

Which estimator to use depends on the assumptions you would like to make about your data. If you think your data should be treated as panel data, with the number of groups growing with the sample size, then even if you have repeated cross-sections, you should use `xtdidregress`.

More in-depth discussions about cluster–robust standard errors can be found in Cameron and Miller (2015) and Wooldridge (2010).

◁

Stored results

didregress stores the following in e():

Scalars
e(N)	number of observations
e(N_clust)	number of clusters
e(tmin)	minimum of first observed treatment time across groups
e(tmax)	maximum of first observed treatment time across groups
e(N_reps)	number of bootstrap replications
e(df_r)	residual degrees of freedom
e(blocksize)	block size used in wild bootstrap computations

Macros
e(cmd)	didregress
e(cmdline)	command as typed
e(depvar)	name of outcome variable
e(treatment)	indicator for treated observations
e(treatment_type)	binary or continuous
e(wtype)	weight type
e(wexp)	weight expression
e(wb_weight)	wild bootstrap error weight
e(datatype)	data type
e(groupvars)	group variables
e(clustvar)	name of cluster variable
e(timevar)	time variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(_contrast_not_ok)	prediction disallowed by contrast

Matrices
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
e(group_count)	matrix with number of groups and treatment time
e(fwboot)	wild bootstrap estimates
e(wboot)	wild bootstrap statistics and constraint
e(aggmethod)	aggregation method

Functions
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices
r(table)	matrix containing the coefficients with their standard errors, test statistics, *p*-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtdidregress` stores the following in `e()`:

Scalars
e(N)	number of observations
e(N_clust)	number of clusters
e(tmin)	minimum of first observed treatment time across groups
e(tmax)	maximum of first observed treatment time across groups
e(N_reps)	number of bootstrap replications
e(df_r)	residual degrees of freedom
e(blocksize)	block size used in wild bootstrap computations

Macros
e(cmd)	xtdidregress
e(cmdline)	command as typed
e(depvar)	name of outcome variable
e(treatment)	indicator for treated observations
e(treatment_type)	binary or continuous
e(wtype)	weight type
e(wexp)	weight expression
e(wb_weight)	wild bootstrap error weight
e(datatype)	data type
e(groupvars)	group variables
e(clustvar)	name of cluster variable
e(panelvar)	panel variable
e(timevar)	time variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(_contrast_not_ok)	prediction disallowed by contrast

Matrices
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
e(group_count)	matrix with number of groups and treatment time
e(fwboot)	wild bootstrap estimates
e(wboot)	wild bootstrap statistics and constraint
e(aggmethod)	aggregation method

Functions
e(sample)	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices
r(table)	matrix containing the coefficients with their standard errors, test statistics, *p*-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

> DID with repeated cross-sectional data
>> DDD model
> DID and DDD models with longitudinal data
> Aggregation estimators
> Wild bootstrap confidence intervals and p-values
> Bias-corrected clustered standard error

DID with repeated cross-sectional data

The DID model for repeated cross-sectional data fit by `didregress` is given by

$$y_{ist} = \gamma_s + \gamma_t + \mathbf{z}_{ist}\beta + D_{st}\delta + \varepsilon_{ist} \tag{1}$$

where i is the observation-level index, s is a group-level index, and t is a time-level index. For example, we might have yearly repeated cross-sectional data for individuals living in different states; in this case, i denotes the individual, s the state, and t the year. In (1) above, γ_s are group fixed effects and γ_t are time fixed effects. $\mathbf{z}_{ist}$ are the covariates, and ε_{ist} is the error term. D_{st} is the treatment that varies at the group and time levels. D_{st} could be binary or continuous.

If y_{ist} is y in the data, then $\mathbf{z}_{ist}$ are z1 and z2, D_{st} is d, the group is `state`, and time is `year`. To fit the model, you would type

```
didregress (y z1 z2) (d), group(state) time(year)
```

which is equivalent to typing

```
areg y z1 z2 i.year d, absorb(state) vce(cluster state)
```

The methods and formulas for this model are those of a linear regression and can be found in *Methods and formulas* in [R] **areg**.

DDD model

The DDD model is given by

$$y_{isgt} = \gamma_s + \gamma_g + \gamma_t + \gamma_s\gamma_t + \gamma_g\gamma_t + \gamma_s\gamma_g + \mathbf{z}_{isgt}\beta + D_{sgt}\delta + \varepsilon_{isgt} \tag{2}$$

where i is the observation-level index, s and g are group-level indices, and t is a time-level index. For example, we might have yearly repeated cross-sectional data for older and younger individuals living in different states. In this case, i denotes the individual, s the state, g age group, and t the year. In (2) above, γ_s are group s fixed effects, γ_g are group g fixed effects, and γ_t are time fixed effects. $\mathbf{z}_{ist}$ are the covariates, and ε_{isgt} is the error term. D_{sgt} denotes the treatment that varies at the group s, group g, and time levels. D_{sgt} could be binary or continuous.

Say you now want to fit a DDD, where the treatment occurs for `group2` in some of the states, `state`. You would type

```
didregress (y z1 z2) (d), group(state group2) time(year)
```

`didregress` would construct the group and time interactions in (2) and fit a linear regression. The methods and formulas for the specification are the same as those in *Methods and formulas* in [R] **areg**.

DID and DDD models with longitudinal data

The DID model for longitudinal data fit by `xtdidregress` is given by

$$y_{ist} = \alpha_i + \gamma_t + \mathbf{z}_{ist}\beta + D_{st}\delta + \varepsilon_{ist} \tag{3}$$

where i is the observation-level index, s is a group-level index, and t is a time-level index. For example, we might have a longitudinal dataset of individuals over time living in a given state; in this case, i denotes the individual, s the state, and t the year. In (3) above, α_i are individual fixed effects and γ_t are time fixed effects. $\mathbf{z}_{ist}$ are the covariates, and ε_{ist} is the error term. D_{st} denotes the treatment that varies at the group and time levels. D_{st} could be binary or continuous. Individuals, i, are assumed to be nested within the group; thus, group effects are subsumed by the individual effects.

If y_{ist} is y in the data, then $\mathbf{z}_{ist}$ are z1 and z2, D_{st} is d, the group is state, time is year, and individuals are denoted by id. To fit the model, you would type

```
xtset id year
xtdidregress (y z1 z2) (d), group(state) time(year)
```

which is equivalent to typing

```
xtreg y z1 z2 i.year d, fe vce(cluster state)
```

The methods and formulas for this model are those of a within estimator computed by xtreg, fe and can be found in *Methods and formulas* in [XT] **xtreg**.

The DDD model is given by

$$y_{isgt} = \alpha_i + \gamma_t + \gamma_t\gamma_s + \gamma_t\gamma_g + \mathbf{z}_{ist}\beta + D_{st}\delta + \varepsilon_{isgt}$$

Aggregation estimators

To discuss the aggregation estimators, it is instructive to rewrite the covariate vector $\mathbf{z}_{ist}$ as consisting of group and time invariant components $\mathbf{z}_{1st}$ and time-, group-, and individual-varying components $\mathbf{z}_{2ist}$. The DID model can now be expressed as

$$y_{ist} = \gamma_s + \gamma_t + \mathbf{z}_{1st}\beta_1 + \mathbf{z}_{2ist}\beta_2 + D_{st}\delta + \varepsilon_{ist}$$
$$y_{ist} = \mathbf{z}_{2ist}\beta_2 + C_{st} + \epsilon_{ist} \tag{4}$$
$$C_{st} = \gamma_s + \gamma_t + \mathbf{z}_{1st}\beta_1 + D_{st}\delta + \nu_{st} \tag{5}$$

The standard and dlang, constant aggregation methods regress y_{ist} on $\mathbf{z}_{2ist}$ in (4) and then obtain an estimate of C_{st}, $\widehat{C}_{st}$. After we regress y_{ist} on $\mathbf{z}_{2ist}$, we get an estimate of the group–time effects $\widehat{C}_{st}$ from $y_{ist} - \mathbf{z}_{2ist}\widehat{\beta}_2 - \widehat{\epsilon}_{ist}$. These effect estimates come from using predict with option d after areg. We then aggregate the data at the s and t levels. With the aggregated data, we run a regression of $\widehat{C}_{st}$ on $\mathbf{z}_{1st}$ and on D_{st} using group and time fixed effects in (5). The dlang method computes the ordinary least-squares standard errors from this second stage. The standard method uses cluster–robust standard errors clustered at the group level by default.

The dlang, varying method runs a regression for each group defined by s and t and obtains an estimate of C_{st} as the set of constants for each regression. Whereas in the first case β_2 was the same vector across the sample, in this case, we will get a different estimate of the slope coefficient for each group. The second step is the same as for dlang, constant.

Wild bootstrap confidence intervals and p-values

The wild cluster bootstrap is a bit different from the conventional bootstrap in that it keeps the covariates fixed and constructs the new dependent variables using the residuals from the original regression. The wild bootstrap procedure proceeds as follows:

1. Fit a restricted model, where the ATET is constrained to be 0. In other words, imposing the null hypothesis that the ATET is 0. Obtain the predicted values, $\widehat{\mathbf{y}}_r$, and residuals, $\widehat{\mathbf{u}}_r$, from the restricted model, where the subscript r refers to the bootstrap replicates. Also fit the unrestricted model and compute the t statistic, t, testing the null hypothesis ATET $= 0$.

2. At each of the subsequent $B - 1$ bootstrap steps, compute $\mathbf{y}^* = \widehat{\mathbf{y}}_r + \widehat{\mathbf{u}}_r \odot (\mathbf{Kw})$, where $\mathbf{w}$ is a column vector of length S, the number of clusters, containing the wild bootstrap weights, and $\mathbf{K}$ is an $N \times S$ matrix with elements $k_{i,j} = 1$ if observation i, $i = 1, \ldots, N$, is in group j and 0 otherwise. The operator $\odot$ is the Hadamard product that performs elementwise multiplication. By default, $w_j = 1$ with probability 0.5 and $w_j = -1$ with probability 0.5, that is, the `rademacher` error weights.

   Alternatively, use one of the following error weights w:

   - `mammen`, which is $1 - \phi$ with probability $\phi/\sqrt{5}$ and ϕ otherwise, where $\phi = (1 + \sqrt{5})/2$.
   - `webb`, which takes the values $-\sqrt{3/2}$, $-\sqrt{2/2}$, $-\sqrt{1/2}$, $\sqrt{1/2}$, $\sqrt{2/2}$, and $\sqrt{3/2}$, each with probability $1/6$.
   - `normal`, which is a normal distribution with the first four moments given by 0, 1, 0, and 3.
   - `gamma`, which is a gamma distribution with shape parameter 4 and scale parameter $1/2$.

3. For each bootstrap sample, compute the unrestricted model and the null hypothesis t statistic, t^*. Include the observed t statistic, t in this set.

4. Given the B bootstrapped t statistics t_b^*, $b = 1, \ldots, B$, compute the wild bootstrap lower-tail p-value

$$P_l(t) = \frac{1}{B} \sum_{b=1}^{B} I(|t| < |t_b^*|)$$

   and the upper-tail p-value

$$P_u(t) = \frac{1}{B} \sum_{b=1}^{B} I(|t| > |t_b^*|)$$

   The reported equal-tail p-value is $P_{et}(t) = 2\min\{P_l(t), P_u(t)\}$.

5. Compute the wild bootstrap confidence interval by searching for the lower limit and upper limit separately using the bisection root-finding algorithm. In doing so, find the quantities ATET $= c_l$ and ATET $= c_u$, $c_l < c_u$, such that $(100 - L)/2\%$ of the bootstrapped distribution of the t statistics testing ATET $= c$ is contained in either the lower or the upper tail. By default, $L = 95$.

   The search algorithm does not always generate an $L\%$ confidence interval. One reason for this situation is if $B(100 - L)/200$ is not an integer, then there is not a slot in the sorted vector $\mathbf{t}^*$ that corresponds to the upper- or lower-tail area $(100 - L)/200$. See Roodman et al. (2019, 8) for a good discussion.

   Also, although a low-probability event, if at least one of the computed $\mathbf{t}^*$ is (numerically) equal to the t testing ATET $= c$, or if there are ties between other computed values in $\mathbf{t}^*$ adjacent to the t testing ATET $= c$, then the algorithm will not achieve the optimal coverage. In this case, we choose the confidence interval that is more conservative than requested. The achieved confidence level is reported in the ATET coefficient table.

   The chance of ties in $\mathbf{t}^*$ increases if the number of clusters, S, is small and the `rademacher` error weights are used, because there are 2^S possible combinations of the two values 1 and -1 in the weight vector $\mathbf{w}$. For example, if $S = 10$, then there are 1,024 possible combinations.

Bias-corrected clustered standard error

Let there be S clusters, $s = 1, \ldots, S$, each with N_s observations, and let N be the number of observations in the data. Let $\mathbf{X}_s$ be the covariate matrix for cluster s, and let $\mathbf{X}$ be the covariate matrix for all observations with dimension $N \times L$. Also, define $\mathbf{I}_{N_s}$ to be an identity matrix of dimension N_s, and define $\mathbf{P}_{ss} = \mathbf{X}_s \left(\mathbf{X}'\mathbf{X} \right)^{-1} \mathbf{X}'_s$ as a projection matrix for cluster s. Finally, let $\widehat{\varepsilon}_s$ be the residuals corresponding to cluster s. The bias-corrected cluster variance–covariance matrix is given by the following:

$$\mathbf{V} = \left(\sum_{s=1}^{S} \mathbf{X}'_s \mathbf{X}_s \right)^{-1} \sum_{s=1}^{S} \mathbf{X}'_s \left(\mathbf{I}_{N_s} - \mathbf{P}_{ss} \right)^{-1/2} \widehat{\varepsilon}_s \widehat{\varepsilon}'_s \left\{ \left(\mathbf{I}_{N_s} - \mathbf{P}_{ss} \right)^{-1/2} \right\}' \mathbf{X}_s \left(\sum_{s=1}^{S} \mathbf{X}'_s \mathbf{X}_s \right)^{-1} \quad (6)$$

Let $\mathbf{P}$ be the projection matrix for all the data, and let $\left(\mathbf{I}_N - \mathbf{P} \right)_s$ be an $N_s \times N$ matrix of rows of the $N \times N$ matrix $\left(\mathbf{I}_N - \mathbf{P} \right)$. Finally, let $\mathbf{e}_{L,k}$ be an L vector with kth component equal to 1 and equal to 0 elsewhere. We define the matrix $\mathbf{G}$ as the matrix with sth column given by

$$\mathbf{G}_s = \left(\mathbf{I}_N - \mathbf{P} \right)'_s \left(\mathbf{I}_{N_s} - \mathbf{P}_{ss} \right)^{-1/2} \mathbf{X}_s \left(\mathbf{X}'\mathbf{X} \right)^{-1} \mathbf{e}_{L,k}$$

We compute the degrees-of-freedom adjustment proposed by Bell and McCaffrey (2002), K_{BM}, as follows,

$$K_{\mathrm{BM}} = \frac{\mathrm{tr}\left(\mathbf{G}'\mathbf{G} \right)^2}{\mathrm{tr}\left\{ \left(\mathbf{G}'\mathbf{G} \right)^2 \right\}}$$

$$= \frac{\left(\sum_{i=1}^{S} \lambda_i \right)^2}{\sum_{i=1}^{S} \lambda_i^2}$$

where $\mathrm{tr}(\cdot)$ is the trace function and λ_i is the ith eigenvalue of the matrix $\mathbf{G}'\mathbf{G}$.

When S is large, computing the eigenvalues can be time consuming. We define

$$\mathbf{a}_s = \left(\mathbf{I}_{N_s} - \mathbf{P}_{ss} \right)^{-\frac{1}{2}} \mathbf{X}_s \left(\mathbf{X}'\mathbf{X} \right)^{-1} \mathbf{e}_{L,k}$$

$$\mathbf{b}_s = \mathbf{P}_s \mathbf{a}_s$$

$$\mathbf{A} = \left(\mathbf{a}'_1 \mathbf{a}_1, \ldots, \mathbf{a}'_S \mathbf{a}_S \right)' = \left(A_1, A_2, \ldots, A_S \right)'$$

$$\mathbf{B} = \left(\mathbf{b}_1, \ldots, \mathbf{b}_S \right) = \begin{pmatrix} B_{1,1} & B_{1,2} & \cdots & B_{1,S} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N,1} & B_{N,2} & \cdots & B_{N,S} \end{pmatrix}$$

where $\mathbf{P}_s = \mathbf{X}\left(\mathbf{X}'\mathbf{X} \right)^{-1} \mathbf{X}'_s$ and $B_{i,s}, i = 1, \ldots, N$, are the elements of the vectors $\mathbf{b}_s, s = 1, \ldots, S$. Then $\mathbf{G}'\mathbf{G} = \mathrm{diag}(\mathbf{A}) - \mathbf{B}'\mathbf{B}$. We now express the adjusted degrees of freedom (Kolesár 2021) as

$$K_{\mathrm{BM}} = \frac{\left(\sum_{s=1}^{S} A_s - \sum_{i=1}^{N} \sum_{s=1}^{S} B_{i,s}^2 \right)^2}{\sum_{s=1}^{S} A_s^2 - 2 \sum_{s=1}^{S} A_s \sum_{i=1}^{N} B_{i,s}^2 + \sum_{s_1=1}^{S} \sum_{s_2=1}^{S} \left(\mathbf{b}'_{s_1} \mathbf{b}_{s_2} \right)^2}$$

which can be computed efficiently in Mata and using QR decomposition. For example, by decomposing $\mathbf{X} = \mathbf{QR}$, where $\mathbf{Q}$ is $n \times k$ and orthonormal and $\mathbf{R}$ is $k \times k$ and upper triangular. The matrix $\mathbf{B}$ can be rewritten so that it has dimension $k \times S$ instead of $N \times S$. We can do even better if the column rank of $\mathbf{X}$ is $r \leq k$ by taking advantage of QR column pivoting.

When weights are specified, we use the weighted-covariate matrix $\widetilde{\mathbf{X}} = \mathrm{diag}(\mathbf{w})^{\frac{1}{2}}\mathbf{X}$ and its corresponding projection matrix $\widetilde{\mathbf{P}}$, as well as the cluster covariance matrices $\widetilde{\mathbf{X}}_s$, their projection matrices $\widetilde{\mathbf{P}}_{ss}$, and weighted residuals $\widetilde{\varepsilon}_s = \mathrm{diag}(\mathbf{w}_s)^{\frac{1}{2}}\widehat{\varepsilon}_s$.

For the ATET coefficient, we use the standard errors from (6) and the degrees of freedom in K_{BM} to construct confidence intervals.

Acknowledgment

References

Abadie, A., and M. D. Cattaneo. 2018. Econometric methods for program evaluation. *Annual Review of Economics* 10: 465–503. https://doi.org/10.1146/annurev-economics-080217-053402.

Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.

———. 2015. *Mastering Metrics: The Path from Cause to Effect*. Princeton, NJ: Princeton University Press.

Bell, R. M., and D. F. McCaffrey. 2002. Bias reduction in standard errors for linear regression with multi-stage samples. *Survey Methodology* 28: 169–181.

Bertrand, M., E. Duflo, and S. Mullainathan. 2004. How much should we trust difference-in-differences estimates? *Quarterly Journal of Economics* 119: 249–275. https://doi.org/10.1162/003355304772839588.

Blundell, R. W., and M. C. Dias. 2009. Alternative approaches to evaluation in empirical microeconomics. *Journal of Human Resources* 44: 565–640. https://doi.org/10.1007/s10258-002-0010-3.

Cameron, A. C., J. B. Gelbach, and D. L. Miller. 2008. Bootstrap-based improvements for inference with clustered errors. *Review of Economics and Statistics* 90: 414–427. https://doi.org/10.1162/rest.90.3.414.

Cameron, A. C., and D. L. Miller. 2015. A practitioner's guide to cluster-robust inference. *Journal of Human Resources* 50: 317–372. https://doi.org/10.3368/jhr.50.2.317.

Clarke, D., and K. Tapia-Schythe. 2021. Implementing the panel event study. *Stata Journal* 21: 853–884.

Donald, S. G., and K. Lang. 2007. Inference with difference-in-differences and other panel data. *Review of Economics and Statistics* 89: 221–233. https://doi.org/10.1162/rest.89.2.221.

Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. https://doi.org/10.1257/jel.47.1.5.

Kolesár, M. 2021. Robust standard errors in small samples. https://cran.r-project.org/web/packages/dfadjust/vignettes/dfadjust.pdf.

Lechner, M. 2011. The estimation of causal effects by difference-in-difference methods. *Foundations and Trends in Econometrics* 4: 165–224. http://doi.org/10.1561/0800000014.

MacKinnon, J. G. 2019. How cluster-robust inference is changing applied econometrics. *Canadian Journal of Economics* 52: 851–881. https://doi.org/10.1111/caje.12388.

MacKinnon, J. G., and M. D. Webb. 2018. The wild bootstrap for few (treated) clusters. *Econometrics Journal* 21: 114–135. https://doi.org/10.1111/ectj.12107.

Moser, P., and A. Voena. 2012. Compulsory licensing: Evidence from the Trading with the Enemy Act. *American Economic Review* 102: 396–427. https://doi.org/10.1257/aer.102.1.396.

Roodman, D., J. G. MacKinnon, M. O. Nielsen, and M. D. Webb. 2019. Fast and wild: Bootstrap inference in Stata using boottest. *Stata Journal* 19: 4–60.

Schmidheiny, K., and S. Siegloch. 2019. On event study designs and distributed-lag models: Equivalence, generalization and practical implications. IZA Discussion Papers No. 12079, Institute of Labor Economics (IZA). https://docs.iza.org/dp12079.pdf.

Wing, C., K. Simon, and R. A. Bello-Gomez. 2018. Designing difference in difference studies: Best practices for public health policy research. *Annual Review of Public Health* 39: 453–469. https://doi.org/10.1146/annurev-publhealth-040617-013507.

Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

[TE] **didregress postestimation** — Postestimation tools for didregress and xtdidregress

[TE] **DID intro** — Introduction to difference-in-differences estimation

[TE] **teffects intro** — Introduction to treatment effects for observational data

[TE] **teffects intro advanced** — Advanced introduction to treatment effects for observational data

[U] **20 Estimation and postestimation commands**