



***The World's Largest Open Access Agricultural & Applied Economics Digital Library***

**This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.**

**Help ensure our sustainability.**

Give to AgEcon Search

AgEcon Search  
<http://ageconsearch.umn.edu>  
[aesearch@umn.edu](mailto:aesearch@umn.edu)

*Papers downloaded from AgEcon Search may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

*No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.*

# From common to firm-specific event dates: A new version of the `estudy` command

Fausto Pacicco

LIUC–Università Carlo Cattaneo  
Castellanza, Italy  
fpacicco@liuc.it

Luigi Vena

LIUC–Università Carlo Cattaneo  
Castellanza, Italy  
lvena@liuc.it

Andrea Venegoni

LIUC–Università Carlo Cattaneo  
Castellanza, Italy  
avenegoni@liuc.it

**Abstract.** The `estudy` command proposed by Pacicco, Vena, and Venegoni (2018, *Stata Journal* 18: 461–476) performs event studies only for event-date clustering, that is, when the event date is common to all securities. This constitutes a relevant limitation because the vast majority of this methodology’s applications concerns studies in which the events happen on different dates for each statistical unit considered. In this article, we propose and describe a substantial update to `estudy`, which 1) performs event studies in the absence of event-date clustering (that is, when each security has its own event date); 2) further customizes the output by producing L<sup>A</sup>T<sub>E</sub>X-formatted tables; 3) graphs the cumulative abnormal returns over a customized period set by the user; 4) makes more output data available through either the return list or Excel files; 5) allows a double possibility as input: either prices or returns; and 6) uses wildcards.

**Keywords:** st0532\_2, `estudy`, event study, financial econometrics

## 1 Introduction

Pacicco, Vena, and Venegoni (2018) presented `estudy`, a community-contributed command to perform an event study. `estudy` allows the user to rely on a flexible framework that can be customized in several ways, from the definition of the estimation window or the event window or windows, to the choice of the statistical model necessary to estimate abnormal returns (ARs), to the parametric and nonparametric statistical tests for their significance.

This contribution remained limited in its applicability because it was suited only to studying events happening on the same date for all the firms in the sample. This is quite binding because the vast majority of event studies’ applications resides in the fields of corporate finance and accounting, where events take place on different dates for each firm. This article aims to fix this lacuna by presenting and describing a substantial update that extends the reach and scope of Pacicco, Vena, and Venegoni (2018) in several ways. More specifically, this version of `estudy` improves on the former because it now allows the user to

- perform event studies for a sample of firms with different event dates, extending the more specific setup (that is, an event study with event-date clustering) proposed in the first release;
- produce L<sup>A</sup>T<sub>E</sub>X-formatted output tables;
- graph the cumulative ARs over a customized period set by the user around the (firm-specific or common) event date;
- retrieve more outputs through either the return list or the Excel file;
- handle securities' and indices' prices as inputs; and
- use wildcards to work with large *varlists*.

## 2 The `estudy` command

The new syntax for `estudy` is

```
estudy varlist1 [ (varlist2) ... (varlistN) ], datevar(varname)
  evdate(string|namelist datelist) dateformat(string) lb1(#) ub1(#)
  [ lb2(#) ub2(#) ... lb6(#) ub6(#) eswlbound(#) eswubound(#)
  modtype(string) indexlist(varlist) diagnosticsstat(string)
  suppress(string) decimal(#) showpvalues nostar outputfile(filename)
  mydataset(datasetname) price tex graph(# # [, save]) detail
```

This article focuses on the new features and changes to Pacicco, Vena, and Venegoni (2018). With respect to omitted options, the reader may refer directly to both Pacicco, Vena, and Venegoni (2018) and the `estudy` help file. As with Pacicco, Vena, and Venegoni (2018), `estudy` still allows the specification of different securities in one to  $N$  *varlists*, separately analyzed. As thoroughly explained by MacKinlay (1997), Kothari and Warner (2007), and Corrado (2011), among others, an event study allows one to estimate the portion of return exclusively determined by the event taking place. It does so by disentangling the ex post actual returns of securities in two parts: the normal return, which is the portion unaffected by the event, and the AR, which measures the impact of the event on securities' returns. In other words, the cumulative AR (CAR) represents the actual net return of securities compared with their normal, expected components. Because this portion of return is taken out from the actual ex post realization, the CAR represents the share of the return solely determined by the event. Using returns as inputs is no longer strictly required, because the user can now perform an event study starting from securities' prices.

## 2.1 Options

As previously mentioned, in this section, we focus on the description of the options that were added or changed in the command presented by Pacicco, Vena, and Venegoni (2018).

`evdate(string| namelist datelist)` specifies the date or dates of the event. For common event dates (that is, event-date clustering), no matter the format of `datevar()`, `evdate()` must be expressed as *mddyyyy*, *ddmmyyyy*, or *yyyymmdd* according to the `dateformat()` specified—MDY, DMY, or YMD, respectively. For different event dates, two variables must be specified: the first is *namelist*, with the names of the securities studied (exactly as reported in *varlists*), and the second is *datelist*, with the dates on which the event takes place for each security. `evdate()` is required.

`outputfile(filename)` specifies the name of the `.xlsx` file in which both the ARs (always without significance stars) and the *p*-values are stored in two separate sheets. The format imposed by `suppress()` is maintained. In the updated version, the test statistics and the standard errors of the CARs<sup>1</sup> are stored in separate sheets. The command automatically replaces the file if it already exists.

`price` has to be specified when the input data are securities' and indices' prices instead of returns. The command then computes the returns and runs the event study as specified. When prices are used, the indices data (`modtype(SIM)`, `modtype(MFM)`, `modtype(MAM)` models) must be inputted as prices.

`tex` shows the output table in `TeX` format.

`graph(# #[, save])` plots the graph ranging from the lower bound (that is, the first integer specified) to the upper bound (that is, the second integer specified). When the option `save` is specified, the graphs will not be shown but will be saved in the working directory. Otherwise, the graphs will be shown but not saved. This option is contingent on the `suppress()` option (when specified); if the individual ARs are suppressed, only group graphs will be created. Contrarily, if the portfolio ARs are suppressed, only individual graphs will be created.

`detail` shows the details of the estimation window length and the other features of the fitted model along with possible warnings that the community-contributed command issued.

## 3 What is new?

The changes made to `estudy` are the following:

1. The main innovation allows users to compute CARs when the event is not clustered on the same date for the securities that are considered. In this case, the user must specify two variables in the `evdate(string| namelist datelist)` option: *namelist*,

---

1. Except for nonparametric tests for which we report the standard deviation of the test statistics.

with the names of the securities studied, and *datelist*, with the dates in which the event takes place for each security. This improvement allows the user to specify as many events as there are companies in *N varlists*. In this respect, notice here that the maximum number of companies that one can simultaneously analyze depends on the Stata version—from about 400 (Stata/IC) to more than 24,000 (for Stata/MP).

2. Specifying the `tex` option produces L<sup>A</sup>T<sub>E</sub>X-formatted output tables in the Results window.
3. By specifying the `graph(# #[, save]`) option, one can graph the cumulative ARs over a user-customized period set around the (firm-specific or common) event date. The suboption `save` allows users to store them in the working directory, while, if it is not specified, the graph will be shown.
4. `estudy` produces more outputs through either the return list or the Excel file. Specifically, it now provides the estimated CARs, as well as their *p*-values and standard deviations (compare footnote 1); the values of the statistical tests adopted; and the time series of estimated ARs. Moreover, `estudy` stores the following in `r()`:

Matrices

<code>r(car)</code>	estimated CARs
<code>r(pv)</code>	<i>p</i> -values of estimated CARs
<code>r(sd)</code>	standard deviation of the test
<code>r(stats)</code>	values of the statistical tests adopted
<code>r(ars)</code>	time series of estimated ARs

6. The `price` option allows users to handle securities' and indices' prices as inputs.
7. The possibility to use wildcards. As *varlists* can be large, we allow users two shortcuts to write them. In particular, users can either put a `-` between two variables or use an `*`. The former includes all the variables between the two specified variables (as defined by the order they are listed in the Variables window); the latter includes all variables starting (or ending) with a given string.

## 4 Examples

We illustrate how `estudy` works using the dataset we provide called `examples_estudy`. This dataset contains the time series of returns and prices of 10 companies' shares and the U.S. S&P 500 Index, as well as the returns of the three Fama and French (1993) factors. Also, the dataset includes two variables that are necessary to perform an event study with multiple event dates: `security_names`, which encompasses the *varnames* specified in all *varlists*,<sup>2</sup> and `event_dates`, which specifies the firm-specific event date corresponding to each security specified in `security_names`. Through the following examples, we show how to use the updated version of the command and clarify how each

2. All *varnames* specified in all *varlists* must be included in the *namelist* argument of `evdate()`. The opposite is not valid, because *namelist* can include more *varnames* than all *varlists*.

option can be used to customize it, with a specific focus on the new or changed components, or both. As previously pointed out, the main change is the possibility to perform an event study without event-date clustering (that is, with multiple, firm-specific event dates); the command Pacicco, Vena, and Venegoni (2018) originally proposed performs event studies on common (single) event dates. Accordingly, this section first shows how to set the command to maintain its original scope (the event study in the case of event-date clustering), and the rest of the discussion is dedicated to an in-depth explanation of the novelties.

Thus, we start with a simple setup by performing an event study on two (separate) *varlists* with three event windows around the event date December 4, 2016. Because we are conducting an event study considering event-date clustering, the *evdate()* option specifies only one argument as string: the common event date. This latter item, coherently with the previous version of the command, must be specified in line with the *dateformat()* option (in this case, MDY, because the event date has the format *mddyyyyy*). The remaining options, unchanged in this update, specify a multifactor model (*modtype()*), the three Fama and French (1993) factors (*indexlist()*), the diagnostic to be implemented (*diagnosticsstat()*), and the number of decimals (*decimal()*) to be shown in the output tables.

```
. use examples_estudy
. estudy ret_ibm ret_cocacola ret_boa ret_ford ret_boeing
> (ret_apple ret.netflix ret_google ret_facebook),
> datevar(date) evdate(12042016) dateformat(MDY)
> modtype(MFM) indexlist(ret_mkt ret_smb ret_hml) diagnosticsstat(BMP)
> lb1(-3) ub1(0) lb2(0) ub2(5) lb3(-3) ub3(3) decimal(4)

Event study with common event date
Event date: 04dec2016, with 3 event windows specified, using the Boehmer,
> Musumeci, Poulsen test


| SECURITY                     | CAAR[-3,0] | CAAR[0,5] | CAAR[-3,3] |
|------------------------------|------------|-----------|------------|
| IBM Corp                     | -1.7104%   | 1.6750%   | -0.6093%   |
| The Coca-Cola Company        | -1.8939%   | 2.7032%   | -0.7428%   |
| Banks of America Corporation | 0.2076%    | 0.6823%   | 0.9800%    |
| Ford Motor Company           | 1.7112%    | 2.6127%   | 4.8299%*   |
| The Boeing Company           | 1.0327%    | -0.4386%  | -0.1827%   |
| Ptf CARs n 1 (5 securities)  | -0.1031%   | 1.4386%   | 0.8787%    |
| CAAR group 1 (5 securities)  | -0.1071%   | 1.4609%** | 0.8876%    |
| Apple Inc                    | 1.4447%    | 1.7658%   | 0.8190%    |
| Netflix Inc                  | 7.4191%    | -1.8969%  | 8.3741%    |
| Alphabet Inc                 | -0.4324%   | 3.7146%   | 1.3000%    |
| Facebook Inc                 | -0.8855%   | 0.8861%   | -0.8564%   |
| Ptf CARs n 2 (4 securities)  | 1.9195%    | 1.1113%   | 2.4423%    |
| CAAR group 2 (4 securities)  | 1.9058%    | 1.1578%   | 2.4573%    |


*** p-value < .01, ** p-value <.05, * p-value <.1
```

As with the original version of the command, when the estimation window is not set, the command considers it, by default, to be from the first available observation to the 30th trading day prior to the event (−30 considering the event date as  $t = 0$ ). A warning message reminds the user of the automatic setup of the estimation window when the

option `detail` is specified. With respect to the output, the table header briefly recaps the setup of the event study performed, recounting the common event date, the number of event windows specified, and the diagnostic test implemented. The first column reports the labels of the variables under scrutiny, adding, by default, two rows per `varlist` to evaluate the impact of the event on the groups of securities. The first shows the CAR obtained through the portfolio approach, and the second shows the cumulative average abnormal return (CAAR). The remaining columns show CARs and CAARs over the event windows specified, identifying those that are statistically significant with asterisks, as explained by the legend at the bottom of the table. Horizontal lines separate the table into panels, with each showing the specified `varlists`. The results yielded by this application tell us that, in the specified event windows, the event that occurred on December 4, 2016, has not exercised a significant impact on all the firms included in the analysis with the sole exception of Ford, which reports a positive and significant AR over the  $[-3, 3]$  window. Moreover, the average CAAR of the first group over the  $[0, 5]$  event window reports a positive and significant AR, showing that the event might have exerted an effect at the aggregate level in the first group. From this, we can infer that some kind of news concerning this company and the group as a whole may have been released on that day. As previously mentioned, the changes implemented to `estudy` make the original command a particular case in this current version. In the next example, we show this by performing an event study on a common event date, which in this case corresponds to December 4, 2016, for all companies under scrutiny. We do so by specifying two arguments in the `evdate()` option: `security_names`, a string variable including at least the `varnames` specified in all `varlists`, and `fs_events`, a date variable that includes the same date (December 4, 2016) for all securities.

SECURITY	EVENT DATE	CAAR[-3,0]	CAAR[0,5]
IBM Corp	04dec2016	-1.7104%	1.6750%
The Coca-Cola Company	04dec2016	-1.8939%	2.7032%
Banks of America Corporation	04dec2016	0.2076%	0.6823%
Ford Motor Company	04dec2016	1.7112%	2.6127%
The Boeing Company	04dec2016	1.0327%	-0.4386%
CAAR group 1 (5 securities)	-	-0.1071%	1.4609%**
Apple Inc	04dec2016	1.4447%	1.7658%
Netflix Inc	04dec2016	7.4191%	-1.8969%
Alphabet Inc	04dec2016	-0.4324%	3.7146%
Facebook Inc	04dec2016	-0.8855%	0.8861%
CAAR group 2 (4 securities)	-	1.9058%	1.1578%

\*\*\* p-value < .01, \*\* p-value <.05, \* p-value <.1

Because we are estimating the impact of a common event using the syntax for firm-specific event dates, the layout changes slightly (the last event window has been removed for the sake of exposition). Indeed, the second column shows the event date referred to for each company, while the portfolio return is no longer computed. Apart from these marginal differences, the reader can easily observe that the results yielded by this framework correspond to those in the previous example.

In the next example, we depart from event-date clustering and investigate companies affected by an event that takes place on different dates for each of them (for example, a dividend announcement, a CEO replacement). We do so by specifying two arguments in the `evdate()` option: the string variable `security_names` and `event_dates`, which is a date variable including the event date corresponding to each security under scrutiny. In addition, rather than specifying each single security, this time we specify more variables thanks to the wildcard `-` (first *varlist*). In line with the command that was originally submitted, the *p*-values are shown in parentheses below each CAAR because the `showpvalues` option has been set.

```

. estudy ret_ibm-ret_boeing
> (ret_apple ret_netflix ret_google ret_facebook),
> datevar(date) evdate(security_names event_dates)
> modtype(SIM) indexlist(ret_sp500) diagnosticsstat(ADJPatell)
> lb1(-3) ub1(3) lb2(-5) ub2(5) decimal(4) showpvalues

Event study with multiple event dates
Event study on multiple event dates, with 2 event windows specified, using the
> Patell test, with the Kolari and Pynnonen adjustment
SECURITY          EVENT DATE   CAAR[-3,3]      CAAR[-5,5]

```

---

IBM Corp	14oct2016	-2.6475% (0.2951)	-3.4600% (0.2759)
The Coca-Cola Company	21oct2016	1.6185% (0.4203)	1.3002% (0.6061)
Banks of America Corporation	12dec2016	0.8476% (0.8112)	0.6169% (0.8899)
Ford Motor Company	01nov2016	-1.4862% (0.6335)	-3.8080% (0.3303)
The Boeing Company	18oct2016	1.6602% (0.5447)	3.1907% (0.3540)
CAAR group 1 (5 securities)	-	0.0229% (0.9560)	-0.3903% (0.8336)
<hr/>			
Apple Inc	14oct2016	1.7534% (0.6488)	3.1514% (0.5144)
Netflix Inc	04nov2016	-5.2361% (0.5476)	-12.9643% (0.2357)
Alphabet Inc	14sep2016	0.8451% (0.7943)	0.3939% (0.9229)
Facebook Inc	11sep2016	1.0014% (0.8668)	3.1636% (0.6818)
CAAR group 2 (4 securities)	-	-0.3304% (0.8858)	-1.4384% (0.9890)
<hr/>			

\*\*\* p-value < .01, \*\* p-value <.05, \* p-value <.1  
p-values in parentheses

Because we are now dealing with firm-specific events, the table header does not show any information regarding the event date but still recalls the implemented statistical test as well as the number of event windows.

Thanks to the wildcard `-`, the first *varlist* includes all variables stored in the dataset from `ret_ibm` to `ret_boeing`. Even though, in this specific case, the wildcard does not allow the user to save much space, its relevance becomes evident. Indeed, using the wildcard offers a twofold benefit: on the one hand, the user can save characters by typing a shorter command, and on the other hand, the user can avoid forgetting variables when the event study is conducted on a large set of securities.

Turning to the results, we see the events have not abnormally affected the returns of companies under scrutiny either individually or as a group. In economic terms, the events considered did not provide additional information to market participants.

The same results can be obtained by using securities' prices instead of returns because the `price` option permits the user to directly handle them, as the next example clarifies.

Event study with multiple event dates			
Event study on multiple event dates, with 2 event windows specified, using the Patell test, with the Kolari and Pynnonen adjustment			
SECURITY	EVENT DATE	CAAR[-3,3]	CAAR[-5,5]
IBM Corp	14oct2016	-2.6475% (0.2951)	-3.4600% (0.2759)
The Coca-Cola Company	21oct2016	1.6185% (0.4203)	1.3002% (0.6061)
Banks of America Corporation	12dec2016	0.8476% (0.8112)	0.6169% (0.8899)
Ford Motor Company	01nov2016	-1.4862% (0.6335)	-3.8080% (0.3303)
The Boeing Company	18oct2016	1.6602% (0.5447)	3.1907% (0.3540)
CAAR group 1 (5 securities)	-	0.0229% (0.9560)	-0.3903% (0.8336)
<hr/>			
Apple Inc	14oct2016	1.7534% (0.6488)	3.1514% (0.5144)
Netflix Inc	04nov2016	-5.2361% (0.5476)	-12.9643% (0.2357)
Alphabet Inc	14sep2016	0.8451% (0.7943)	0.3939% (0.9229)
Facebook Inc	11sep2016	1.0014% (0.8668)	3.1636% (0.6818)
CAAR group 2 (4 securities)	-	-0.3304% (0.8858)	-1.4384% (0.9890)
<hr/>			

\*\*\* p-value < .01, \*\* p-value <.05, \* p-value <.1  
p-values in parentheses

We can easily observe that the results are unchanged; the only difference between the previous example and this one lies in the input data: the former relies on returns, while the latter relies on security prices (as emerges from the option `price`).

A new option made available with this release allows users to further customize the output format. Along these lines, the `tex` option permits users to obtain L<sup>A</sup>T<sub>E</sub>X-formatted tables, which can be readily integrated into the T<sub>E</sub>X files. Below, we provide a concrete example by performing an event study with two *varlists* and two event windows over four trading days ( $[-3, 0]$ ) and six trading days ( $[0, 5]$ ).

```
. estudy pr_ibm-pr_boeing
> (pr_apple pr_netflix pr_google pr_facebook),
> datevar(date) evdate(security_names event_dates)
> modtype(HMM) indexlist(pr_sp500) price diagnosticsstat(KP)
> lb1(-3) ub1(0) lb2(0) ub2(5) decimal(4) tex

Event study with multiple event dates
\title{Event study on multiple event dates, with 2 event windows specified,
> using the Boehmer, Musumeci, Poulsen test, with the Kolari and Pynnonen
> adjustment}
SECURITY & EVENT DATE & CAAR[-3,0] & CAAR[0,5] \\
\midrule
IBM Corp & 14oct2016 & -1.6385\% & -2.6790\% \\
The Coca-Cola Company & 21oct2016 & 1.1572\% & 0.5500\% \\
Banks of America Corporation & 12dec2016 & 1.6497\% & -3.2187\% \\
Ford Motor Company & 01nov2016 & -2.3816\% & -2.3639\% \\
The Boeing Company & 18oct2016 & 1.2081\% & 3.3201\% \\
CAAR group 1 (5 securities) & - & 0.0157\% & -0.8425\% \\
\midrule
Apple Inc & 14oct2016 & 1.1188\% & -0.6756\% \\
Netflix Inc & 04nov2016 & -3.0507\% & -7.3402\% \\
Alphabet Inc & 14sep2016 & -1.8584\% & 1.5903\% \\
Facebook Inc & 11sep2016 & -2.3840\% & 0.9783\% \\
CAAR group 2 (4 securities) & - & -1.4977\% & -1.2875\% \\
\midrule
\caption{*** p-value < .01, ** p-value <.05, * p-value <.1}
```

Still, the events do not seem to provide new information, because none of the CARs and CAARs reach statistical significance.

The last example is dedicated to the explanation of the `graph()` option. We thus show how to obtain the graphs of estimated CARs and CAARs over the customized range specified through the option  $(-20, +20)$ . In other words, in the  $t$ th period, the graph shows the value of CAARs from the lower bound specified in the `graph()` option to period  $t$ . Accordingly, when the length of the event window is equal to the one in the graph, the last point of the graph corresponds to the value of the CAARs reported in the output table (the last column in this case).

```

. estudy ret_ibm-ret_amazon,
> datevar(date) evdate(security_names event_dates)
> modtype(HMM) indexlist(ret_mkt ret_smb ret_hml) diagnosticsstat(KP)
> suppress(ind) lb1(-3) ub1(0) lb2(-20) ub2(20)
> decimal(4) graph(-20 20)

Event study with multiple event dates
Event study on multiple event dates, with 2 event windows specified, using the
> Boehmer, Musumeci, Poulsen test, with the Kolari and Pynnonen adjustment
SECURITY          EVENT DATE  CAAR[-3,0]          CAAR[-20,20]
-----
CAAR group 1 (10 securities)      -      -0.5283%      1.3882%
-----
*** p-value < .01, ** p-value <.05, * p-value <.1

```

Because the `suppress(ind)` option has been set, `estudy` shows the corresponding graph on a group basis only.

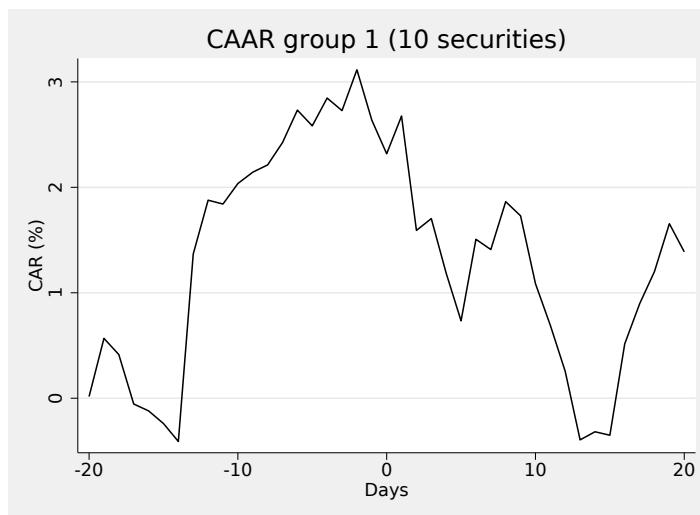


Figure 1. The graph shows the CAAR in the specified period, which ranges from 20 days prior to the event to 20 days after the event

If the user specifies the suboption `save` of the `graph()` option, `estudy` does not show any graphs but rather stores them in the directory in use.

## 5 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```

. net sj 21-1
. net install st0532.2      (to install program files, if available)
. net get st0532.2         (to install ancillary files, if available)

```

## 6 References

- Corrado, C. J. 2011. Event studies: A methodology review. *Accounting & Finance* 51: 207–234. <https://doi.org/10.1111/j.1467-629X.2010.00375.x>.
- Fama, E. F., and K. R. French. 1993. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics* 33: 3–56. [https://doi.org/10.1016/0304-405X\(93\)90023-5](https://doi.org/10.1016/0304-405X(93)90023-5).
- Kothari, S. P., and J. B. Warner. 2007. Econometrics of event studies. In *Handbook of Empirical Corporate Finance*, vol. 1, ed. B. E. Eckbo, 3–36. Amsterdam: Elsevier.
- MacKinlay, A. C. 1997. Event studies in economics and finance. *Journal of Economic Literature* 35: 13–39.
- Pacicco, F., L. Vena, and A. Venegoni. 2018. Event study estimations using Stata: The estudy command. *Stata Journal* 18: 461–476. <https://doi.org/10.1177/1536867X1801800211>.

### About the authors

Fausto Pacicco is an assistant professor of economics at LIUC–Università Carlo Cattaneo, from which he received his PhD. His research interests are time-series analysis, microeconomics, territorial marketing, and competitive advantage; he is also involved in the development of microterritorial statistic platforms. He teaches public economics, political economy, and time-series analysis. He has published several articles in journals such as the *Journal of Empirical Finance*, *Journal of International Money and Finance*, *Stata Journal*, and *Annals of Tourism Research*.

Luigi Vena is assistant professor of finance at LIUC–Università Carlo Cattaneo, from which he received a PhD in management, finance, and legal disciplines. He teaches the theory of financial intermediation, corporate governance, international financial markets, and financial systems. His research interests include macroprudential supervision of financial systems, monetary policy, corporate governance, and corporate disclosure. He has published several articles in journals such as the *Journal of Financial Stability*, *Journal of Empirical Finance*, *Journal of International Money and Finance*, *Journal of Cleaner Production*, *Stata Journal*, and *Journal of International Financial Management & Accounting*.

Andrea Venegoni is assistant professor of economics at LIUC–Università Carlo Cattaneo, from which he received a PhD in management, finance, and legal disciplines. His research interests include monetary economics and policy, empirical macroeconomics, macroprudential supervision of financial systems, and financial market functioning. He has published several articles in journals such as the *Journal of Financial Stability*, *Journal of Empirical Finance*, *Journal of International Money and Finance*, and *Stata Journal*. He lectures on macroeconomics, microeconomics, economic policy, and econometrics at LIUC–Università Carlo Cattaneo.