The Stata Journal (2024)
24, Number 3, pp. 478–502 DOI: 10.1177/1536867X241276112

cnevent: Event study with Chinese equity market data

Chuntao Li
School of Economics
Henan University
Kaifeng, China
chtl@henu.edu.cn

Yizhuo Fang
School of Economics
Henan University
Kaifeng, China
yzhfang1@163.com

Lifang Cao
School of Economics
Henan University
Kaifeng, China
caolifang64@163.com

Abstract. In this article, we present a new command, `cnevent`, that runs event studies about Chinese-listed companies. With `cnevent`, researchers are required to provide only a list of events with the Chinese stock code and the corresponding date for each event, and the command can automatically extract indexes and each individual stock's return data to run the whole process of the event study. Furthermore, `cnevent` enables users to choose the benchmark from among different market indexes and different event window sets with options. The command then generates daily abnormal returns for all trading days within the event window and aggregates the cumulative abnormal returns (CARs) for the whole event window. Finally, `cnevent` can plot a graph to show the trend of the CAR_t within the event window and test whether the event has a significant effect on valuation.

Keywords: st0754, cnevent, Chinese listed companies, event study, abnormal returns, cumulative abnormal returns, CARs

1 Introduction

Event studies are commonly used empirical analysis methods developed by finance researchers. They are designed to measure the impact of an event on stock returns or test the extent to which the stock price reacts to information disclosure (MacKinlay 1997; Kolari, Pape, and Pynnonen 2018). Their objective is to evaluate the degree to which investors earn abnormal returns (ARs) due to an event that carries new information. In this article, an AR means the difference between the observed return and the expected return without the specific event, which is predicted by estimation models.

Examples of events are a merger or acquisition announcement (Zhang 2007), an earnings announcement (Raghavendra and Vermaelen 1998), the passage of a new legislation document (Iliev 2010), or the change of corporate governance practice (Lin et al. 2019). Since the establishment of the Chinese stock market, many events have occurred, and more rules have been introduced to the market, which creates a good test ground for financial theories with event study methodology.

In a classical event study, we need to collect event lists, set the event and estimation windows, predict the (cumulative) AR using an estimation model, and conduct statistical tests. However, achieving the above steps is time consuming and tedious because it takes a good deal of manual work, including collecting daily stock return data within the estimation window according to the calendar date, fitting the regression model within the estimation window, and calculating ARs and cumulative abnormal returns (CARs) within the event window. Of course, some of the steps can be achieved easily by community-contributed commands such as `estudy` (Pacicco, Vena, and Venegoni 2018), `eventstudy` (Zhang, Li, and Xu 2013), and `eventstudy2` (Kaspereit 2015), which have full functionality for estimation models, statistical tests, and output results. Nevertheless, these commands have different characteristics and applications (Kaspereit 2021). For example, `eventstudy` is a basic command that cannot test hypotheses, but it does well in the graphical user interface; `estudy` complements the hypothesis-testing function and presents the output in various formats, including LaTeX, Excel, and so on; `eventstudy2` provides the most comprehensive benchmark models and test statistics. Additionally, it is suitable for dealing with events with lacking data, infrequent trading, and large samples.

As for the process of data input, the above commands are complex because they require the collection of multiple datasets (the event list, the trading data of individual stocks, and the market or factor returns) and specify the applicative format of these datasets. Pacicco, Vena, and Venegoni (2018) indicate that, when using the command `estudy`, one is required to specify a dataset that contains the time series of market returns and individual stock returns. Similarly, `eventstudy` and `eventstudy2` also require the datasets of security trading. (Zhang, Li, and Xu 2013; Kaspereit 2015). Clearly, such requirements can be challenging for some researchers.

To simplify the data input process, we introduce a new command, `cnevent`, that can perform an event study about Chinese companies without investing loads of time in collecting data. The command operates based on another command, `cntrade` (Zhang and Li 2014)[1], that enables users to easily access the data of Chinese-listed companies from the Cloud. Thus, users need to provide only a list of events with the stock code and the corresponding event date for each event, and then the command extracts indices and daily individual stock return data to perform the entire process of an event study.

In general, varied test statistics, abundant benchmark models, and superior output are desirable features of event study software (Kaspereit 2021). `cnevent` can provide common test statistics (nonparametric and parametric tests) and benchmark models (capital asset pricing model [CAPM], constant mean return model, and market-adjusted model). For the presentation of output, `cnevent` can generate the dataset of ARs and CARs of the sample companies within the specified event window. It displays the cumulative average abnormal returns (AARs) in tabular and graphical formats and provides the tabulation of statistical test results. Additionally, `cnevent` provides a set of options to set the estimation window, the event window, and the market benchmark with different indexes.

1. `cntrade` downloads historical market quotations for a list of Chinese stock codes or index codes from NetEase (a website providing financial information in China; see https://money.163.com). The command was later updated to get data from the China East Money website (https://www.eastmoney.com) because of a change of the website structure of NetEase.

All in all, `cnevent` provides basic functions comprehensively, which helps inexperienced users easily complete event studies about Chinese companies. Although the variety of statistical tests and benchmark models of `cnevent` is less than that of `eventstudy2`, the former has great advantages in automatic data collection and simple operation, which is helpful for newcomers to carry out event studies or inexperienced Stata users. Additionally, we try to obtain the list of stock codes and announcement dates according to Bailey, Huang, and Yang (2011) and use `cnevent` to output the results of ARs, CARs, and parametric and nonparametric tests with their specified windows. The concise commands run smoothly and receive similar results.

This article is structured as follows. In section 2, we introduce the steps of an event study. In section 3, we describe the `cnevent` command in more detail, including a brief overview of the command and definitions for the parameters. We also describe the syntax that we have designed along with its definitions, and we describe the options available for the command. In section 4, we illustrate the whole process of an event study with some examples. In section 5, we conclude and make suggestions for further refinement and extensions of the command.

2 The event study steps

Although event studies can be designed and conducted differently, they all follow explicit methodologies and steps. Normally, the following five steps outline the whole procedure for an event study.

1. **Find a list of events**

   The first step is to find a relevant list, which is a two-dimensional table containing two variables. One variable is the stock code, or the company ticker in the US market. The second variable is the corresponding event date. The event must be some shock that may cause an extraordinary market reaction. Specifically, the company's stock price associated with the event, or even the entire stock market, may undergo an excessive change because of the event. Usually, it is a new piece of legislation or a change of trading rules with firm-specific events, which generate the same event date for all companies. For the event study, researchers may check each company's information to collect the event list. The list should be complete to avoid selection bias. Past research projects have used the earnings announcements (Kross and Schroeder 1984) or mergers and acquisitions (MA) announcements (Halpern 1983) of each company as the events, so the event dates for each company are different. If we are studying the effect of an MA on 100 companies, the list needs to have 100 observations representing 100 MA companies. Corresponding event dates here may be the first announcement date of the MA for each firm.

2. **Define the event window**

   In this step, we define the event window as a time interval around the event date, usually ranging from a few days before to a few days after the event date. This means that an event will affect the company's stock returns over a period of time around the event date. We can observe the company's returns within the event window to see whether anything unusual happens. It is worth noting that in event studies, the event window starts before the event occurs because there may be information leakage in the financial market. The event window may be defined as $(-2, 3)$, which means from two trading days before the event to three trading days after the event, so this window contains six trading days, including the event date. We also need to emphasize here that we are talking about trading days instead of calendar days.

3. **Define the estimation window and select an estimation model**

   In the third step, we need to define the estimation window and select an estimation model to compute the normal or expected returns. The estimation window is a sample period prior to the event window, usually leaving several days between the two windows to exclude the interference from the event itself. Normally, researchers need to choose the length of the estimation window. When the length of the estimation window is wide, the sampling error of the market model almost disappears (MacKinlay 1997), so a long estimation period is more common, although there might be potential event clustering (Mckenzie, Thomsen, and Dixon 2004). In contrast, a short period may avoid the effect of event clustering. Meanwhile, a slightly worse test statistic performance of a narrow estimation window indicates that the shorter estimation window should be considered when the parameter of the benchmark model is not stable (Corrado and Zivney 1992). Typical lengths of the estimation window range from 100 to 300 days for daily studies (Peterson 1989). The estimation window, defined as $(-200, -60)$, means from 200 trading days before the event to 60 trading days before the event. This window contains 141 trading days.[2]

   Several models are available for estimating normal or expected stock returns, including the CAPM, the market-adjusted model (naïve model), the constant mean return model, and the Fama–French three-factor (or four-factor) model. The CAPM is widely used as a benchmark in event studies because it offers a simple and efficient way to measure the expected return. It assumes that investors require higher returns for more risk, and the risk of an investment can be measured through its beta. Several studies have investigated the applicability of the CAPM in event studies and have confirmed it as a suitable benchmark. For example, Fama and French (1992) found that the CAPM was a useful tool for measuring ARs in event studies. Studies by Sharpe (1964) and Lintner (1965) also support using the CAPM as a benchmark in event studies.

2. Note that we are specifically talking about trading days, not calendar days.

4. **ARs and CARs**

In this step, ARs are calculated by the difference between real returns and returns predicted by an estimation model within the event window. According to Sorokina, Booth, and Thornton (2013), we can calculate ARs with the following equation if we select the CAPM as our estimation model:

$$\mathrm{AR}_{i,t} = R_{i,t} - \left(\widehat{\alpha_i} + \widehat{\beta_i} R_{m,t}\right) \tag{1}$$

In (1), $\mathrm{AR}_{i,t}$ is the AR of a stock i at time t within the event window, $R_{i,t}$ is the actual return of a stock i at time t, $\widehat{\beta_i}$ is the market-risk coefficient, and $R_{m,t}$ is the market return at time t. $(\widehat{\alpha_i} + \widehat{\beta_i} R_{m,t})$ is the expected return of a stock i, which is determined by $\widehat{\alpha_i}$, $\widehat{\beta_i}$, and $R_{m,t}$. We estimate the $\widehat{\alpha_i}$ and $\widehat{\beta_i}$ by regressing the stock return $R_{i,t}$ against the market return $R_{m,t}$ within the estimation window. Incidentally, the closing price is a traditional indicator used to estimate the capital asset price volatility (Garman and Klass 1980), and it is often used to calculate daily actual returns (Woolridge 1983; Armitage 1995), especially in event studies (MacKinlay 1997).

Owing to Brown and Warner (1980), we provide two other useful and relatively simple models: the constant mean return model and the market-adjusted model. The AR can be calculated by (2) and (3), respectively:

$$\mathrm{AR}_{i,t} = R_{i,t} - R_{m,t} \tag{2}$$

$$\mathrm{AR}_{i,t} = R_{i,t} - \overline{R}_i \tag{3}$$

In (2), $\mathrm{AR}_{i,t}$ is the AR of a stock i at time t within the event window, $R_{i,t}$ is the actual return of a stock i at time t, and $R_{m,t}$ is the market return at time t. In (3), $\overline{R}_i$ is the average return of a stock i within the estimation window.

Then we ensure that the predicted errors for ARs both across firms and time are aggregated. For the most part, aggregating across firms involves an averaging of ARs for all firms in the sample on a given date t. The AARs can be calculated by (4):

$$\mathrm{AAR}_t = \frac{1}{N} \sum_{i}^{N} \mathrm{AR}_{i,t} \tag{4}$$

Aggregating across time is the accumulation of multiday ARs within the event window for a firm i. The CARs are described by (5):

$$\mathrm{CAR}_i(t_1, t_2) = \sum_{t=t_1}^{t_2} \mathrm{AR}_{i,t} \tag{5}$$

In (4) and (5), $\mathrm{AR}_{i,t}$ is the AR of a stock i at time t, AAR_t is the mean of a pool of firms at time t, and $\mathrm{CAR}_i(t_1, t_2)$ is the sum of $\mathrm{AR}_{i,t}$ from time t_1 to time t_2.

5. Statistical tests

Event studies are concerned with whether there are ARs on or around an event date. Thus, the CARs should be statistically tested in the last step of the event study to answer this question. The hypothesis testing on event studies covers a wide range of tests that can be classified into parametric and nonparametric tests. Parametric tests (at least in the field of event studies) assume that the individual firm's ARs are normally distributed, whereas nonparametric tests do not rely on such assumptions.

Several parametric tests are available for significance tests, including Student's t test, the cross-sectional test, the Patell test, the adjusted Patell test, the standardized cross-sectional test, and the adjusted standardized cross-sectional test. Some common nonparametric tests are the sign test and the Wilcoxon signed-rank test.

Student's t test is widely used as a parametric test in event studies because it is a simple and efficient way to test the AR. Armitage (1995) thinks that most event studies are concerned with the effect of some event or investment rule on many stocks, so it is usually AARs (errors) for one or more portfolios of N stocks that are calculated and tested for significance by Student's t test. For example, Brown and Warner (1980) use Student's t test to assess the performance of a simple methodology based on the market model. McGuire and Dilts (2008) examine the economic value of successive generations of the ISO 9000 standard by Student's t test. So we add Student's t test in the `cnevent` package to test the CARs.

However, there are several implicit shortcomings of Student's t test when used to assess abnormal performance: for example, when you construct the t-test statistic, security returns must be normally distributed. If such assumptions were not met, the assumed sampling distribution could differ from the actual distribution, which would result in false inferences. According to MacKinlay (1997), we consider the nonparametric tests with less restrictive assumptions than the t test to ensure the effectiveness of significance tests. Specifically, we also use the Wilcoxon signed-rank test used in actual event studies.

3 The cnevent command

3.1 Overview

As mentioned above, to run `cnevent`, you must first install the external command `cntrade` (Zhang and Li 2014) from the Statistical Software Components (SSC) Archive. Second, note that the command is compatible only with Stata 16 or above. Third, users must load a list of events into memory. Once these conditions are met, `cnevent` can calculate the ARs and the CARs for each event based on the estimation model. The list of events in memory is as follows:

stkcd	edate
2	2014-04-14
600900	2015-04-14
600000	2016-03-14
601898	2018-05-21
601988	2013-02-05
300999	2021-01-17

The list of events usually contains two variables: the stock code (`stkcd`) that identifies the subject of each sample company and the corresponding event date (`edate`) when the event occurs. `cnevent` can automatically identify different event date formats and allow users to convert them to either string or Stata date format. If an event occurred on a holiday date, `cnevent` automatically regards the first trading day after the holiday as the event date for an event study.

3.2 Syntax

The basic syntax for `cnevent` is

`cnevent` *varlist* [, `eventw(`*numlist*`)` `estw(`*numlist*`)` `model(`*string*`)` `ar(`*string*`)`
`car(`*string*`)` `index(`*int*`)` `estsmpn(`*int*`)` `filename(`*string*`)`
`graph(`*filename*[, *type*]`)` `t(`*filename*`)` `wilcoxon(`*filename*`)`]

The *varlist* contains two variables, the stock code and the event date, that should be separated by spaces. The command also has some customized options that are discussed in detail in the next section.

3.3 Options

The following options can be used with `cnevent`.

`eventw(`*numlist*`)` sets the event window. The default is `eventw(-3,5)`, which specifies an event window from the third trading day before the event date to the fifth trading day after the event date. Multiple event windows can be set if needed. For example, event windows could be $[-4,4]$, $[-3,5]$, $[-2,6]$, and $[-1,7]$, specified by `eventw(-4,4 -3,5 -2,6 -1,7)`.

`estw(`*numlist*`)` sets the estimation window. The default is `estw(-200,-10)`, which specifies an estimation window that starts from the 200th trading day before the event date and ends at the 10th trading day before the event date. If the estimation window was overlapped with the event window, the command would stop running.

`model(`*string*`)` specifies the estimation model used to calculate ARs. The default is `model(1)`, which represents the CAPM. `model(2)` represents the constant mean return model, and `model(3)` represents the market-adjusted model.

`ar(`*string*`)` sets an output variable name header that represents ARs. The default is `ar(AR)`.

`car(`*string*`)` sets an output variable name header that represents CARs. The default is `car(CAR)`.

`index(`*int*`)` sets an index as the market benchmark returns. The default is `index(300)`, which stands for China Securities Index 300 (CSI 300). Similarly to the S&P 500, CSI 300 tracks a basket of 300 listed blue chip companies in the Shanghai and Shenzhen stock exchanges. There are also other alternative indexes. Some of the commonly used index codes in China's equity market are shown in table 1.

Table 1. Market indexes

Index	Description
000001	Shanghai composite index
000002	Shanghai A-share composite index
000003	Shanghai B-share composite index
000300	CSI 300
399001	Shenzhen component index
399003	Shenzhen B-share component index
399005	Shenzhen small- and medium-sized 100-firm index
399006	Shenzhen growth enterprise market index
399008	Shenzhen small- and medium-sized 300-firm index

`estsmpn(`*int*`)` restricts the minimum trading days for each firm within 365 days prior to the event. The default is `estsmpn(50)`. If there are fewer than 50 trading days in the 365 days before the event date, the event study for the underlying event will not be carried out because of insufficient samples.

`filename(`*string*`)` sets the name of the output `.dta` file in which the ARs and CARs are saved. The default is `filename(CAR)` with extension `.dta`.

`graph(`*filename*[`,` *type*]`)` requests that `cnevent` plot the average trend of CAR_t within the event window and save the graph with the filename specified. The format of the output file is specified by the suboption *type*, whose default format is `gph`. The commonly used output types are shown in table 2.

Table 2. Types of graph

Short form	Full form
`gph`	Stata graph format
`ps`	Postscript
`eps`	Encapsulated postscript
`svg`	Scalable vector graphics
`pdf`	Portable document format
`png`	Portable network graphics

`t(`*filename*`)` performs the two-tail Student's t test and produces an output `.docx` file to save the t-test table.

`wilcoxon(`*filename*`)` performs the Wilcoxon signed-rank test and produces an output `.docx` file to save the signed-rank test table.

4 Examples

We provide some examples to illustrate the whole process of `cnevent` and conduct the event study. We show the designated syntax for `cnevent` and clarify how each option available for the command is customized to meet the users' requirements.

Before we proceed, `cntrade` (Zhang and Li 2014) must be installed from SSC with the following command:

```
. ssc install cntrade, replace
```

Suppose we have a series of events that can be input to Stata with the following command:

```
. input stkcd str10 edate
            stkcd       edate
  1. 000002 "2014-04-14"
  2. 600900 "2015-04-14"
  3. 600000 "2016-03-14"
  4. 601898 "2018-05-21"
  5. 601988 "2013-02-05"
  6. 300999 "2021-01-17"
  7. 301308 "2021-03-07"
  8. end
```

We start with the default setup by running the command without any options. In this case, our event window and estimation window are set as $[-3, 5]$ and $[-200, -10]$, respectively; the market index is CSI 300; and the estimation model is CAPM. Furthermore, within 365 calendar days, the minimum number of trading days before the event date is 50. With the default setting, the two output variables have the name headers of AR and CAR. Also, the output file is named CAR. The `graph()`, `t()`, and `wilcoxon()` options are not used with default settings, so there will be no graph output or statistical test. The command to type is as follows:

```
. cnevent stkcd edate
```

Then `cnevent` automatically extracts data from the Cloud and calculates the ARs and CARs. When Stata runs `cnevent`, the output is displayed in the Results window, where the stock code, the event date, and a pair of numbers indicating how many events out of the total have been done are reported, so users are informed of the whole process. Additionally, `cnevent` can display and remove the company with the event date prior to the initial public offering date. The output is as follows:

```
current working on Stock  000002 with event date 2014-04-14, 1 of 7 completed
current working on Stock  600900 with event date 2015-04-14, 2 of 7 completed
current working on Stock  600000 with event date 2016-03-14, 3 of 7 completed
current working on Stock  601898 with event date 2018-05-21, 4 of 7 completed
current working on Stock  601988 with event date 2013-02-05, 5 of 7 completed
current working on Stock  300999 with event date 2021-01-17, 6 of 7 completed
Your eventdate of 301308 lies before the IPO date
```

At the end of the program, the event study dataset that includes the firm's ID (`stkcd`), the event date (`edate`), the ARs (`AR*`), and the CARs (`CAR*`) is generated. Here the character `n` after `AR` in the variable name means a negative number; for example, `ARn3` means AR(-3), or the AR 3 trading days before the event date. Similarly, `CARn35` means CAR$(-3, 5)$, the CAR of an event window running from 3 trading days before to 5 days after the event date. One can use the `describe` command to examine the variables in memory. The variable labels are self-explanatory.

```
. describe
Contains data from CAR.dta
 Observations:            6
    Variables:           12                  20 Nov 2022 12:44
--------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
--------------------------------------------------------------------------------
stkcd           float   %9.0g
edate           float   %dCY-N-D
ARn3            float   %9.0g                 AR(-3)
ARn2            float   %9.0g                 AR(-2)
ARn1            float   %9.0g                 AR(-1)
AR0             float   %9.0g                 AR(0)
AR1             float   %9.0g                 AR(1)
AR2             float   %9.0g                 AR(2)
AR3             float   %9.0g                 AR(3)
AR4             float   %9.0g                 AR(4)
AR5             float   %9.0g                 AR(5)
CARn35          float   %9.0g                 CAR[-3,5]
--------------------------------------------------------------------------------
Sorted by:
```

The output below is a dataset resulting from the above command `cnevent`. The columns `ARn3`–`ARn1` represent the ARs from the third to the first trading day before the event date. The column `AR0` represents the ARs on the event date. The columns `AR1`–`AR5` represent the ARs from the first to the fifth trading day after the event date, respectively. The column `CARn35` represents the CARs within the event window $[-3, 5]$.

```
. list stkcd edate ARn3 ARn2 ARn1 AR0
```

	stkcd	edate	ARn3	ARn2	ARn1	AR0
1.	2	2014-04-14	-.0127015	-.0196959	-.010246	-.0009848
2.	600900	2015-04-14	-.0042335	-.0064629	.0152437	.0188737
3.	600000	2016-03-14	.0655268	.0061188	-.082931	-.0186572
4.	601898	2018-05-21	.0127835	-.0032913	.0246431	-.0051835
5.	601988	2013-02-05	.0035393	.0098117	.0154173	-.0210583
6.	300999	2021-01-17	-.1076018	-.0374199	-.0175819	-.0539155

```
. list AR1 AR2 AR3 AR4 AR5 CARn35
```

	AR1	AR2	AR3	AR4	AR5	CARn35
1.	.0010707	.0127763	-.0103626	.0043929	.006304	-.029447
2.	-.0141367	.0322055	.0050057	-.004954	.0310461	.0725875
3.	.0058938	.0543472	-.02736	-.0039879	-.0057149	-.0067644
4.	.0010598	-.0127347	-.0036787	-.0039982	.0030039	.0126039
5.	-.0035225	-.0203266	.0020352	.0006587	-.003886	-.0173311
6.	.0431716	.049007	-.0470141	-.0741183	.0257646	-.2197081

The variable names of the ARs and CARs can be specified further by the `ar()` and `car()` options. In addition, the `filename()` option sets the name of the `.dta` file, which stores the output data. The command to type is as follows:

```
. cnevent stkcd edate, ar(AR_k) car(CAR_k) filename(eventstudy)
```

With the above options, nothing is changed except the variable names of AR and CAR, as indicated below.

```
. list stkcd edate AR_kn3 AR_kn2 AR_kn1 AR_k0
```

	stkcd	edate	AR_kn3	AR_kn2	AR_kn1	AR_k0
1.	2	2014-04-14	-.0127015	-.0196959	-.010246	-.0009848
2.	600900	2015-04-14	-.0042335	-.0064629	.0152437	.0188737
3.	600000	2016-03-14	.0655268	.0061188	-.082931	-.0186572
4.	601898	2018-05-21	.0127835	-.0032913	.0246431	-.0051835
5.	601988	2013-02-05	.0035393	.0098117	.0154173	-.0210583
6.	300999	2021-01-17	-.1076018	-.0374199	-.0175819	-.0539155

```
. list AR_k1 AR_k2 AR_k3 AR_k4 AR_k5 CAR_kn35
```

	AR_k1	AR_k2	AR_k3	AR_k4	AR_k5	CAR_kn35
1.	.0010707	.0127763	-.0103626	.0043929	.006304	-.029447
2.	-.0141367	.0322055	.0050057	-.004954	.0310461	.0725875
3.	.0058938	.0543472	-.02736	-.0039879	-.0057149	-.0067644
4.	.0010598	-.0127347	-.0036787	-.0039982	.0030039	.0126039
5.	-.0035225	-.0203266	.0020352	.0006587	-.003886	-.0173311
6.	.0431716	.049007	-.0470141	-.0741183	.0257646	-.2197081

In event studies, it is typical to fit the CAPM in an estimation window within 365 calendar days prior to the event date. In some occasions, trading days could be few because of trading suspension or a new initial public offering. This may bias the estimation of β_i and α_i because there is a small sample. We use the option `estsmpn(200)` to require the trading days to be no fewer than 200 days during this one-year period. Otherwise, the program will omit the corresponding event by reporting the ARs and CAR data as missing values. The following `cnevent` command requires that there should be no fewer than 100 trading days within the 365 calendar days prior to the event date.

```
. cnevent stkcd edate, estsmpn(100)
  (output omitted)
. list stkcd edate ARn3 ARn2 ARn1 AR0
```

	stkcd	edate	ARn3	ARn2	ARn1	AR0
1.	2	2014-04-14	-.0127015	-.0196959	-.010246	-.0009848
2.	600900	2015-04-14	-.0042335	-.0064629	.0152437	.0188737
3.	600000	2016-03-14	.0655268	.0061188	-.082931	-.0186572
4.	601898	2018-05-21	.0127835	-.0032913	.0246431	-.0051835
5.	601988	2013-02-05	.0035393	.0098117	.0154173	-.0210583
6.	300999	2021-01-17	.	.	.	.

```
. list AR1 AR2 AR3 AR4 AR5 CARn35
```

	AR1	AR2	AR3	AR4	AR5	CARn35
1.	.0010707	.0127763	-.0103626	.0043929	.006304	-.029447
2.	-.0141367	.0322055	.0050057	-.004954	.0310461	.0725875
3.	.0058938	.0543472	-.02736	-.0039879	-.0057149	-.0067644
4.	.0010598	-.0127347	-.0036787	-.0039982	.0030039	.0126039
5.	-.0035225	-.0203266	.0020352	.0006587	-.003886	-.0173311
6.	.	.	.	.	.	.

Usually, ARs can be calculated by several different models. The `model()` option can specify the estimation model. The following `cnevent` command uses the constant mean return model, whose code is 2.

```
. cnevent stkcd edate, estsmpn(100) model(2)
  (output omitted)
. list stkcd edate ARn3 ARn2 ARn1 AR0
```

	stkcd	edate	ARn3	ARn2	ARn1	AR0
1.	2	2014-04-14	-.0113073	.0009927	-.0114073	-.0015073
2.	600900	2015-04-14	-.0128649	.0057351	.0261351	.0192351
3.	600000	2016-03-14	.0617471	.0031471	-.0808529	-.0071529
4.	601898	2018-05-21	.0049372	-.0105628	.0341372	-.0008628
5.	601988	2013-02-05	.0032859	.0160859	.0158859	-.0185141
6.	300999	2021-01-17	.	.	.	.

```
. list AR1 AR2 AR3 AR4 AR5 CARn35
```

	AR1	AR2	AR3	AR4	AR5	CARn35
1.	-.0204073	.0151927	-.0142073	.0048927	-.0144073	-.052166
2.	-.0267649	.0529351	.0164351	-.0199649	.0449351	.1058157
3.	.0088471	.0586471	-.0189529	.0072471	.0116471	.0443241
4.	-.0027628	-.0255628	-.0106628	-.0068628	.0070372	-.0111654
5.	-.0031141	-.0221141	.0032859	-.0031141	-.0096141	-.0179272
6.	.	.	.	.	.	.

Furthermore, there are many different indexes in the Chinese stock market. The `index()` option can specify which market index is to be used as the market benchmark return. The following command uses the Shanghai A-share composite index, whose code is 000002. Here we allow users to omit zeros in the code; `cnevent` automatically adds the zeros from the front when the code specifies fewer than six characters.

```
. cnevent stkcd edate, estsmpn(100) index(2)
  (output omitted)
. list stkcd edate ARn3 ARn2 ARn1 AR0
```

	stkcd	edate	ARn3	ARn2	ARn1	AR0
1.	2	2014-04-14	-.0163169	-.0189649	-.0091564	-.0025307
2.	600900	2015-04-14	-.0025938	-.0071575	.0114607	.0192202
3.	600000	2016-03-14	.0648855	.0062219	-.082995	-.0191563
4.	601898	2018-05-21	.0143534	-.0042552	.0171983	-.0096924
5.	601988	2013-02-05	.0028226	.0107219	.0144349	-.0193193
6.	300999	2021-01-17	.	.	.	.

```
. list AR1 AR2 AR3 AR4 AR5 CARn35
```

	AR1	AR2	AR3	AR4	AR5	CARn35
1.	-.0006457	.0124609	-.010248	.0052929	.0069202	-.0331885
2.	-.013983	.0336442	.0011128	-.0041054	.0329334	.0705317
3.	.0068958	.0563777	-.0274571	-.0046291	-.0029648	-.0028212
4.	-.0032129	-.0066858	-.0047606	-.0013661	.0095605	.0111392
5.	-.0033114	-.0196142	.0011131	-.0014119	-.0036191	-.0181834
6.	.	.	.	.	.	.

`cnevent` allows users to choose different event windows or estimation windows with options. The `eventw()` option is used to set multiple event windows, and the `estw()` option can be used for adjusting the estimation window. Although multiple event windows can be used at the same time, note that only one estimation window can be set. The following example uses the three event windows $[-1, 3]$, $[-1, 4]$, and $[-1, 5]$, which are set by the `eventw()` option. It also uses one estimation window, $[-190, -10]$, which is set by the `estw()` option.

```
. cnevent stkcd edate, estsmpn(100) eventw(-1,3 -1,4 -1,5) estw(-190,-10)
  (output omitted)
. list stkcd edate ARn1 AR0 AR1 AR2
```

	stkcd	edate	ARn1	AR0	AR1	AR2
1.	2	2014-04-14	-.0099146	-.0006338	.0007768	.0132178
2.	600900	2015-04-14	.0151563	.0188104	-.0141703	.0320954
3.	600000	2016-03-14	-.0825663	-.0181973	.0062673	.0547343
4.	601898	2018-05-21	.0243618	-.0052865	.0012374	-.0122467
5.	601988	2013-02-05	.0153813	-.0211144	-.003558	-.0203408
6.	300999	2021-01-17	.	.	.	.

```
. list AR3 AR4 AR5 CARn13 CARn14 CARn15
```

	AR3	AR4	AR5	CARn13	CARn14	CARn15
1.	-.0101138	.0047754	.0060337	-.0066677	-.0018922	.0041415
2.	.004917	-.0049821	.0309518	.0568088	.0518268	.0827786
3.	-.0269314	-.0035307	-.0051959	-.0666935	-.0702242	-.0754201
4.	-.0033921	-.0038536	.0029108	.0046739	.0008202	.003731
5.	.0019916	.0006638	-.003862	-.0276402	-.0269764	-.0308383
6.	.	.	.	.	.	.

The `graph()` option can plot a graph to show the trend of the CAR_t within the range specified by `eventw()`. Though we may set different event windows, there is only one graph for CAR_t, where the time t ranges from the lower bound to the upper bound of the event windows. In figure 1, the time line ranges from -5 to 5, although we set multiple event windows by `eventw(-5,1 -2,2 -1,5)`. The filename is specified by the graph() option. Furthermore, the suboption *type* can be used to set the format of the graph. For instance, we can output the graph to a .pdf document. The command to type is as follows:

```
. cnevent stkcd edate, estsmpn(100) eventw(-5,1 -2,2 -1,5)
>    estw(-190,-10) graph(p1, pdf)
```

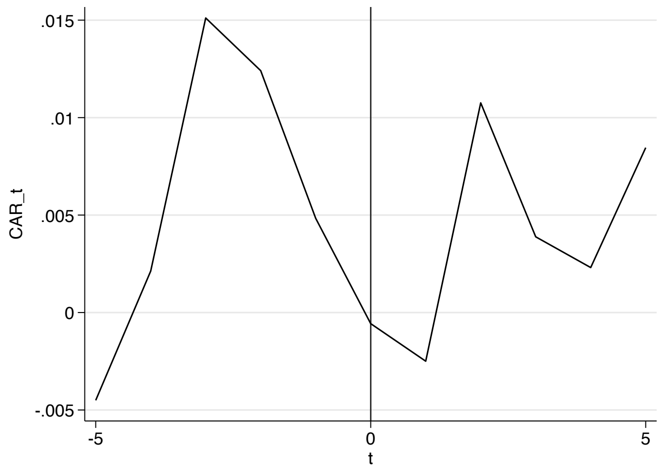

Figure 1. Dynamics of CAR_t

Finally, users may let cnevent perform additional Student's t and Wilcoxon signed-rank tests to see whether the CARs across the event window are significantly different using the t() and wilcoxon() options. The output is not only displayed on the Results window but also exported to an external .docx file, with the filename specified by the t() and wilcoxon() options. In table 3 (table 4), the first column presents all the event windows that have been set. The second column presents the mean (median) values of CARs within different event windows. The last column presents the statistical values for the corresponding Student's t test (Wilcoxon signed-rank test). The command to type is as follows:

```
. cnevent stkcd edate, estsmpn(100) eventw(-5,1 -2,2 -1,5)
>    estw(-190,-10) graph(p1) t(t1) wilcoxon(w1)
```

Table 3. Student's t-test table

	Mean	t
$(-5, 1)$	-0.0025	-0.1270
$(-2, 2)$	-0.0044	-0.3104
$(-1, 5)$	-0.0039	-0.1494

Table 4. Wilcoxon signed-rank test table

	Median	z
$(-5, 1)$	0.0102	0
$(-2, 2)$	-0.0171	0.5443
$(-1, 5)$	0.0030	-0.2722

Furthermore, suppose we have a common event date when a new piece of legislation is passed. We intend to study the impact of the new legislation on Shanghai A-share listed companies. We first need to use the external command `cnstock`[3] to find the list of all A-share companies listed on the Shanghai Stock Exchange (we need to install `cnstock` from SSC). Here we eliminate specially treated firms (whose stock name is labeled with `ST`) and randomly select 100 companies from the list of all Shanghai A-share listed companies. The common event date is January 23, 2020. We can type the following commands:

```
. ssc install cnstock
. cnstock SHA
. drop if index(stknm, "ST")
. set seed 100
. sample 100, count
. generate edate = "2020-1-23"
```

After obtaining the list of events, we use `cnevent` to perform the event study. In this case, the estimation window is set as $[-190, -10]$, the event window is set as $[-2, 3]$, and the market index is CSI 300. Within 365 calendar days, the minimum number of trading days before the event date is set to be 100, and the generated file is labeled as `CommonEvent`. Other settings use their defaults; the AR is labeled as AR, and the CAR is labeled as CAR. Then we use the `summarize` command to produce the summary statistics of the ARs and CARs. The commands are as follows:

3. `cnstock` (Li, Li, and Xue 2016) is another community-contributed command to collect the latest stock codes in China, which downloads stock names and stock codes of China's listed companies from https://quote.cfi.cn/stockList.aspx.

```
. cnevent stkcd edate,estw(-190,-10) eventw(-2,3) index(300)
>     estsmpn(100) filename(CommonEvent)
```

(*output omitted*)

```
. summarize AR* CAR*

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
        ARn2 |         60    .0075616    .0262668  -.0798094   .0841094
        ARn1 |         60    .0012968    .0249136  -.0724543     .09241
         AR0 |         60    .0001876    .0241177  -.0820352   .0722014
         AR1 |         60    -.015302    .0454997  -.2642057   .1229368
         AR2 |         60   -.0252046    .0526364  -.1337629    .154598
-------------+---------------------------------------------------------
         AR3 |         60    .0086806    .0334625  -.0706813   .1800035
      CARn23 |         60     -.02278     .088086  -.1738526   .2561711
```

Because we set up only one event window, the results may not be as expected. To efficiently study the impact of the event, we set multiple event windows to observe the results. We use nine event windows ($[-2,1]$, $[-2,2]$, $[-2,3]$, $[-1,1]$, $[-1,2]$, $[-1,3]$, $[0,1]$, $[0,2]$, and $[0,3]$), which are set by the `eventw()` option. The commands are as follows:

```
. cnevent stkcd edate, estw(-190,-10) index(300) estsmpn(100)
>     eventw(-2,1 -2,2 -2,3 -1,1 -1,2 -1,3 0,1 0,2 0,3)
>     filename(CommonEvent1)
```

(*output omitted*)

```
. summarize AR* CAR*

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
        ARn2 |         60    .0075616    .0262668  -.0798094   .0841094
        ARn1 |         60    .0012968    .0249136  -.0724543     .09241
         AR0 |         60    .0001876    .0241177  -.0820352   .0722014
         AR1 |         60    -.015302    .0454997  -.2642057   .1229368
         AR2 |         60   -.0252046    .0526364  -.1337629    .154598
-------------+---------------------------------------------------------
         AR3 |         60    .0086806    .0334625  -.0706813   .1800035
      CARn21 |         60    -.006256     .089677  -.4380401   .2695247
      CARn22 |         60   -.0314606    .0954132  -.2834422   .2373129
      CARn23 |         60     -.02278     .088086  -.1738526   .2561711
      CARn11 |         60   -.0138176    .0733843  -.3810415   .2069613
-------------+---------------------------------------------------------
      CARn12 |         60   -.0390222    .0812924   -.231243   .1957119
      CARn13 |         60   -.0303416    .0750879  -.1853385   .2218633
       CAR01 |         60   -.0151143    .0617756   -.346241   .1593903
       CAR02 |         60    -.040319    .0698212  -.2118014   .2005655
       CAR03 |         60   -.0316383    .0677852  -.1658969    .226717
```

In addition, some firms may not have enough trading days within 365 calendar days before the event, which makes the estimation unreliable. Thus, we use the option `estsmpn(200)` to require the number of trading days to be no fewer than 200 during this one-year period. The commands are as follows:

```
. cnevent stkcd edate, estw(-190,-10) index(300) estsmpn(200)
>     eventw(-2,1 -2,2 -2,3 -1,1 -1,2 -1,3 0,1 0,2 0,3)
>     filename(CommonEvent2)
```

(*output omitted*)

```
. summarize AR* CAR*
```

Variable	Obs	Mean	Std. dev.	Min	Max
ARn2	59	.0086558	.025075	-.0798094	.0841094
ARn1	59	.0019086	.0246687	-.0724543	.09241
AR0	59	.0015812	.0217521	-.053873	.0722014
AR1	59	-.0110833	.031932	-.0599653	.1229368
AR2	59	-.0282521	.0474502	-.1337629	.0759254
AR3	59	.0057769	.0249882	-.0706813	.0749599
CARn21	59	.0010624	.0700835	-.1545866	.2695247
CARn22	59	-.0271898	.0902627	-.2197572	.2373129
CARn23	59	-.0214129	.0881978	-.1738526	.2561711
CARn11	59	-.0075934	.0557999	-.1193087	.2069613
CARn12	59	-.0358456	.0781441	-.231243	.1957119
CARn13	59	-.0300687	.0757025	-.1853385	.2218633
CAR01	59	-.009502	.0442668	-.0998671	.1593903
CAR02	59	-.0377542	.0675096	-.2118014	.2005655
CAR03	59	-.0319773	.0683157	-.1658969	.226717

Based on the summary statistics above, the mean values for all `CAR*` variables have negative signs except `CARn21`, which suggests that the event that happened within the specified event window had a negative effect on Shanghai A-share-listed companies.

Next we use the option `graph()` to plot a graph (figure 2) to show the trend of the CAR_t within multiple event windows.

```
. cnevent stkcd edate, estw(-190,-10) index(300) estsmpn(200)
>     eventw(-2,1 -2,2 -2,3 -1,1 -1,2 -1,3 0,1 0,2 0,3)
>     filename(CommonEvent2) graph(CommonEvent2)
```

(*output omitted*)

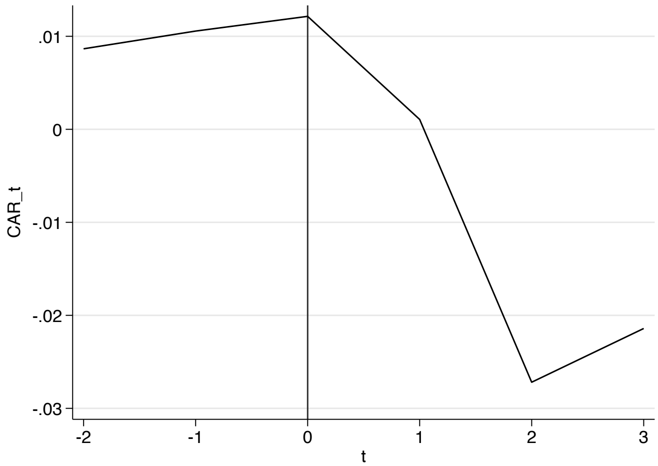

Figure 2. Dynamics of CAR_t

Finally, with an additional `t()` option, we examine whether the event has a significant effect on valuation by the two-tail Student's t test. The corresponding results are reported in a `.docx` file with the filename specified by the option. For example, the following code will report the t-test results in `CommonEvent2.docx` and present them on the screen. The format of the output is shown in table 5.

```
. cnevent stkcd edate, estw(-190,-10) index(300) estsmpn(200)
>    eventw(-2,1 -2,2 -2,3 -1,1 -1,2 -1,3 0,1 0,2 0,3)
>    filename(CommonEvent2) graph(CommonEvent2) t(CommonEvent2)
  (output omitted)
```

Table 5. Student's t-test table

	Mean	t
$(-2, 1)$	0.0011	0.1164
$(-2, 2)$	−0.0272	−2.3138
$(-2, 3)$	−0.0214	−1.8649
$(-1, 1)$	−0.0076	−1.0453
$(-1, 2)$	−0.0358	−3.5234
$(-1, 3)$	−0.0301	−3.0509
$(0, 1)$	−0.0095	−1.6488
$(0, 2)$	−0.0378	−4.2956
$(0, 3)$	−0.0320	−3.5954

We also intend to examine the impact of the common event on the Chinese financial market. As an example, we randomly select 100 companies from the list of all Chinese-listed companies. Our event windows are set as $[-2,1]$, $[-2,2]$, $[-2,3]$, $[-1,1]$, $[-1,2]$, $[-1,3]$, $[0,1]$, $[0,2]$ and $[0,3]$; the estimation window is set as $[-190,-10]$; and the market index is CSI 300. We use the option `estsmpn(200)` to require the number of trading days to be no fewer than 200 during a one-year period; we use the options `graph()` and `t()` to plot a graph to show the trend of the CAR_t within the event windows and test whether the event has a significant effect on valuation. Finally, we use `summarize` to produce summary statistics of ARs and CARs. The commands are as follows with the graph displayed as figure 3 and the format of the output shown in table 6.

```
. cnstock all
  (output omitted)
. drop if index(stknm, "ST")
  (output omitted)
. generate edate = "2020-1-23"
. set seed 100
. sample 100, count
  (output omitted)
. cnevent stkcd edate, estw(-190,-10) index(300) estsmpn(200)
>     eventw(-2,1 -2,2 -2,3 -1,1 -1,2 -1,3 0,1 0,2 0,3)
>     filename(SCommonEvent2) graph(SCommonEvent2) t(SCommonEvent2)
  (output omitted)
. summarize AR* CAR*
```

Variable	Obs	Mean	Std. dev.	Min	Max
ARn2	63	.0108103	.0343697	-.037716	.125308
ARn1	63	-.0024216	.0317008	-.1070068	.0966189
AR0	63	.001873	.0316599	-.0682519	.1410775
AR1	63	.0040601	.058372	-.0969284	.20195
AR2	63	-.0344885	.0511738	-.1156585	.0702174
AR3	63	.0153384	.0317816	-.0706813	.0905507
CARn21	63	.0143218	.1090988	-.1959406	.5624388
CARn22	63	-.0201667	.1296572	-.2222895	.5296986
CARn23	63	-.0048282	.1403398	-.1950996	.6172617
CARn11	63	.0035115	.0834333	-.1582246	.4396464
CARn12	63	-.030977	.1058	-.1845735	.4069062
CARn13	63	-.0156385	.1168981	-.1785875	.4944693
CAR01	63	.0059331	.0777759	-.1360279	.3430275
CAR02	63	-.0285554	.1002296	-.1772846	.3102874
CAR03	63	-.013217	.1136082	-.1620752	.3978505

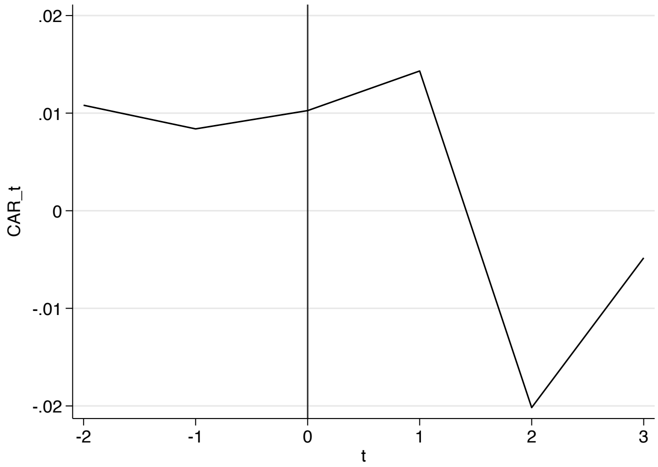

Figure 3. Dynamics of CAR_t

Table 6. Student's t-test table

	Mean	t
$(-2, 1)$	0.0143	1.0420
$(-2, 2)$	-0.0202	-1.2345
$(-2, 3)$	-0.0048	-0.2731
$(-1, 1)$	0.0035	0.3341
$(-1, 2)$	-0.0310	-2.3239
$(-1, 3)$	-0.0156	-1.0618
$(0, 1)$	0.0059	0.6055
$(0, 2)$	-0.0286	-2.2613
$(0, 3)$	-0.0132	-0.9234

5 Conclusion

Our command, `cnevent`, makes performing event studies for Chinese companies more convenient. Users are required to provide only a list of events with the stock code and the event date and choose the estimation and event windows with options. After that, the command will complete the whole process, including collecting data from public financial websites and reporting results for the event study.

Compared with other commands for event studies, the limitations here are twofold: first, `cnevent` can perform event studies only in China because stocks' daily return data are downloaded from the Cloud database of Chinese-listed companies; second, `cnevent` introduces several common benchmark models and statistical tests but not all existing models and tests.

In the future, as more and more trading information from different countries becomes available online, we hope to update the command to cover more countries. We also have a plan to encompass more models and tests, including the Fama–French three-factor (or four-factor) model and other statistical tests (like the Patell z statistic, the adjusted Patell statistic, and the Boehmer–Musumeci–Poulsen test).

6 Programs and supplemental material

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

```
. net sj 24-3
. net install st0754      (to install program files, if available)
. net get st0754          (to install ancillary files, if available)
```

The command `cnevent` can be installed from the SSC archive by typing

```
. ssc install cnevent
```

7 References

Armitage, S. 1995. Event study methods and evidence on their performance. *Journal of Economic Surveys* 9: 25–52. https://doi.org/10.1111/j.1467-6419.1995.tb00109.x.

Bailey, W., W. Huang, and Z. S. Yang. 2011. Bank loans with Chinese characteristics: Some evidence on inside debt in a state-controlled banking system. *Journal of Financial and Quantitative Analysis* 46: 1795–1830. https://doi.org/10.1017/S0022109011000433.

Brown, S. J., and J. B. Warner. 1980. Measuring security price performance. *Journal of Financial Economics* 8: 205–258. https://doi.org/10.1016/0304-405X(80)90002-1.

Corrado, C. J., and T. L. Zivney. 1992. The specification and power of the sign test in event study hypothesis tests using daily stock returns. *Journal of Financial and Quantitative Analysis* 27: 465–478. https://doi.org/10.2307/2331331.

Fama, E. F., and K. R. French. 1992. The cross-section of expected stock returns. *Journal of Finance* 47: 427–465. https://doi.org/10.2307/2329112.

Garman, M. B., and M. J. Klass. 1980. On the estimation of security price volatilities from historical data. *Journal of Business* 53: 67–78. https://doi.org/10.1086/296072.

Halpern, P. 1983. Corporate acquisitions: A theory of special cases? A review of event studies applied to acquisitions. *Journal of Finance* 38: 297–317. https://doi.org/10.2307/2327962.

Iliev, P. 2010. The effect of SOX Section 404: Costs, earnings quality, and stock prices. *Journal of Finance* 65: 1163–1196. https://doi.org/10.1111/j.1540-6261.2010.01564.x.

Kaspereit, T. 2015. eventstudy2: Stata module to perform event studies with complex test statistics. Statistical Software Components S458086, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s458086.html.

———. 2021. Event studies with daily stock returns in Stata: Which command to use? *Stata Journal* 21: 462–497. https://doi.org/10.1177/1536867X211025835.

Kolari, J. W., B. Pape, and S. Pynnonen. 2018. Event study testing with cross-sectional correlation due to partially overlapping event windows. Research Paper 3167271, Mays Business School. https://doi.org/10.2139/ssrn.3167271.

Kross, W., and D. A. Schroeder. 1984. An empirical investigation of the effect of quarterly earnings announcement timing on stock returns. *Journal of Accounting Research* 22: 153–176. https://doi.org/10.2307/2490706.

Li, C., Z. Li, and Y. Xue. 2016. cnstock: Stata module to download stock names and codes for Chinese listed companies. Statistical Software Components S458271, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s458271.html.

Lin, C., M. S. Officer, T. Schmid, and H. Zou. 2019. Is skin in the game a game changer? Evidence from mandatory changes of D&O insurance policies. *Journal of Accounting and Economics* 68: art. 101225. https://doi.org/10.1016/j.jacceco.2019.01.001.

Lintner, J. 1965. Security prices, risk, and maximal gains from diversification. *Journal of Finance* 20: 587–615. https://doi.org/10.2307/2977249.

MacKinlay, A. C. 1997. Event studies in economics and finance. *Journal of Economic Literature* 35: 13–39.

McGuire, S. J., and D. M. Dilts. 2008. The financial impact of standard stringency: An event study of successive generations of the ISO 9000 standard. *International Journal of Production Economics* 113: 3–22. https://doi.org/10.1016/j.ijpe.2007.02.045.

Mckenzie, A. M., M. R. Thomsen, and B. L. Dixon. 2004. The performance of event study approaches using daily commodity futures returns. *Journal of Futures Markets* 24: 533–555. https://doi.org/10.1002/fut.10126.

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.

Peterson, P. P. 1989. Event studies: A review of issues and methodology. *Quarterly Journal of Business and Economics* 28: 36–66.

Raghavendra, R. P., and T. Vermaelen. 1998. Glamour, value and the post-acquisition performance of acquiring firms. *Journal of Financial Economics* 49: 223–253. https://doi.org/10.1016/S0304-405X(98)00023-3.

Sharpe, W. F. 1964. Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance* 19: 425–442. https://doi.org/10.1111/j.1540-6261.1964.tb02865.x.

Sorokina, N., D. E. Booth, and J. H. Thornton, Jr. 2013. Robust methods in event studies: Empirical evidence and theoretical implications. *Journal of Data Science* 11: 575–606. https://doi.org/10.6339/JDS.2013.11(3).1166.

Woolridge, J. R. 1983. Dividend changes and security prices. *Journal of Finance* 38: 1607–1615. https://doi.org/10.2307/2327590.

Zhang, I. X. 2007. Economic consequences of the Sarbanes–Oxley Act of 2002. *Journal of Accounting and Economics* 44: 74–115. https://doi.org/10.1016/j.jacceco.2007.02.002.

Zhang, X., and C. Li. 2014. Importing Chinese historical stock market quotations from NetEase. *Stata Journal* 14: 381–388. https://doi.org/10.1177/1536867X1401400209.

Zhang, X., C. Li, and X. Xu. 2013. eventstudy: Stata module to perform event studies in finance. Statistical Software Components S457615, Department of Economics, Boston College. https://econpapers.repec.org/software/bocbocode/s457615.htm.

About the authors

Chuntao Li is a professor of finance at Henan University in Kaifeng, China.

Yizhuo Fang is a PhD candidate of finance at Henan University in Kaifeng, China.

Lifang Cao is a PhD candidate of finance at Henan University in Kaifeng, China.