Instructions for Generating Analyst-Based Peer Groups with R Using the peergroups Function

Markku Kaustia
Aalto University

Ville Rantala
University of Miami

January 12, 2017

1. General description of the code

The code consists of an R function that uses simulation to produce analyst-based peer groups as defined in Kaustia and Rantala (2017). The input data can be from any source and the time periods that are used as inputs do not necessarily have to be years. The output data consists of time-period-specific firm - peer firm pairs that identify the analyst-based peers separately for each input firm.

The user can adjust the confidence interval and the number of simulation rounds, if required. The code can also be used to produce peer groups from other types of data with similar characteristics: The observations do not necessarily have to consist of analysts or firms.

2. Usage

2.1. Calling the function

The function is named peergroups and you can call it as follows:

```
peergroups(timeindex, firmindex, analystindex)
```

The three required arguments of the function are vectors `timeindex`, `firmindex`, and `analystindex`. The input data must consist of rows of observations that identify time-period-specific firm-analyst pairs (see example on the next page). Each observation must have three components: time index (such as year), firm index (firm identifier, such as IBES Ticker, PERMNO, or GVKEY), and analyst index (analyst identifier such as IBES analyst code). A single observation indicates that a specific analyst (identified by analyst index) follows a specific firm (identified by firm index) during a specific time period (identified by time index).

The three input vectors must be formed so that `timeindex` contains the time index values of all the observations, `firmindex` contains the firm index values of all the observations, and `analystindex` contains the analyst index values of all the observation so that the same element in each vector refers to the same observation. For example, if the *n*th element in vector `timeindex` is year1, the *n*th element in vector `firmindex` is firm2 and the *n*th element in vector `analystindex` is analyst3, then the *n*th observation indicates that firm2 is followed by analyst3 in year1.

Here is an example of possible input data

timeindex	**firmindex**	**analystindex**
year1	firm1	analyst1
year1	firm1	analyst2
year1	firm2	analyst1
year1	firm2	analyst4
year1	firm2	analyst5
year2	firm1	analyst1
year2	firm1	analyst6
year2	firm2	analyst1
year2	firm2	analyst2
year2	firm2	analyst8
year2	firm3	analyst9
year2	firm3	analyst10

In the example above, there are two firms in year1 and three firms in year2. analyst1 follows firm1 and firm2 during both years, and there are no other common analysts between the firms.

The observations do not have to be sorted in any particular way, and the identifiers do not have to be of any particular data type. The function will automatically remove any duplicate observations. The input vectors should not contain any NA values (missing observations).

2.2. Optional arguments

The function has two optional arguments: `interval` and `repetitions`.

`interval` defines the confidence interval used in the simulation. The default value is 0.99. If C_i denotes the peer criterion of firm *i*, an `interval` of 0.99 indicates that the probability that firm *i* has C_i common analysts with another random firm is less than 1% (i.e. 1 - `interval`). Possible values for this argument must be between 0 and 1.

`repetitions` gives the number of simulation rounds used in the simulation. The default value is 1,000.

2.3. Function output

When the function is run, the percentage of processed firm-year observations will periodically be printed on the R console to provide information about the progress of the code.

The function returns a data frame with five columns. The names of the columns are `time_index, firm, peer_firm, common_analysts, and peer_criterion.` Each row in the data frame consists of a firm - peer firm pair in a specific time period. `time_index` is the time period, `firm` is the identifier of a firm, and `peer_firm` is the identifier of its analyst-based peer firm. `peer_criterion` reports the value of the peer criterion for the firm (the minimum number of common analysts it must have with a peer firm), and `common_analysts` is the actual number of common analysts between the two firms during the time period.

Here is an example of what possible output could look like

time_index	**firm**	**peer_firm**	**common_analysts**	**peer_criterion**
year1	firm1	firm2	5	3
year1	firm1	firm3	4	3
year1	firm2	firm1	5	4
year2	firm1	firm2	4	3
year2	firm1	firm3	6	3
year2	firm2	firm4	7	5

In the example above, firm1 and firm2 are each other's peers in year1. In year1, the peer group of firm1 consists of firm2 and firm3, and the peer group of firm2 consists of a single firm (firm1). In year2, firm2 is firm1's peer, but firm1 is not firm2's peer.

Note that there can be non-mutual peer relationships, where firm A is firm B's peer but firm B is not firm A's peer. Also note that if firms A and B are both each other's peers, the output data will contain a row identifying firm A as firm B's peer and another row identifying firm B as firm A's peer.

3. Additional tips

Here are some additional tips:

- If you want to exclude certain analyst-firm observations from the data, remember to do it before running the code. In Kaustia and Rantala (2017) we exclude analyst codes that are associated with more than 50 different firms in a single year.
- With large datasets, the code can take many hours to run. Increasing the number of repetitions will also increase the processing time. If you wish to speed up the simulation and have access to multiple processing cores, you can divide the input data into subsamples consisting of separate time periods and run the code with parallel processing.

- The function will periodically print the percentage of processed firm-year observations on the R console together with a time stamp. The difference between individual time stamps allows you to estimate the remaining processing time.
- If you use IBES Tickers as company identifiers, note that one company in IBES has ticker "NA". If you have it in your data, make sure that it is stored as a String, because the missing value NA will cause an error in the code.
- Before running the code with full data, it is usually a good idea to use a smaller sample to test that it functions as desired with your input vectors.

4. General notes about the code

R is not the fastest available programming language for this type of simulation, but we provide the general purpose code in R because of its ease of use and adaptability. The same simulation can easily be coded with other programming languages following the description in Kaustia and Rantala (2017), if needed.

The code is provided "as is" and we assume no liability whatsoever for the results generated by the code or for any possible effects resulting from its use.

References

Kaustia, M., & Rantala, V. (2017). Common analyst –method for defining peer firms. Working paper.