The Stata Journal (2018)
18, Number 2, pp. 387–394

Attrition diagrams for clinical trials and meta-analyses in Stata

Christiaan H. Righolt
Vaccine and Drug Evaluation Centre
Department of Community Health Sciences
University of Manitoba
Winnipeg, Canada
Christiaan.Righolt@umanitoba.ca

Salaheddin M. Mahmud
Vaccine and Drug Evaluation Centre
Department of Community Health Sciences
University of Manitoba
Winnipeg, Canada
Salah.Mahmud@umanitoba.ca

Abstract. In this article, we present `attrition`, a suite of commands to simplify the maintenance and documentation of implemented exclusion criteria and attrition conditions using standard Stata facilities and to generate an attrition diagram. `attrition` can be used, both from the command line and in do-files, to keep the diagram up to date with the analysis it documents. Six subcommands (`set`, `exclude`, `count`, `tab`, `list`, `graph`) allow the diagram to be constructed in a versatile way.

Keywords: st0527, attrition set, attrition exclude, attrition count, attrition tab, attrition list, attrition graph, attrition diagram, inclusion variable, clinical trial, meta-analysis

1 Introduction

Clinical investigators often restrict participation in clinical trials and prospective observational epidemiological studies to those who meet predefined inclusion and exclusion criteria. For instance, participants diagnosed with a certain disease at the enrollment visit might be excluded from the trial for patient safety or treatment appropriateness reasons. Additionally, exclusions, commonly termed "attrition", are sometimes performed retrospectively after data collection to reduce bias, for example, to exclude participants who did not adhere to study protocol or were lost to follow-up. This is often done by flagging records that meet all criteria and limiting the records used in a particular analysis to the flagged participants.

Exclusions and attrition can result in biased estimates, especially if they result in a much smaller study group (say, <90% of the original count), and may limit the generalizability of the study findings (Fergusson et al. 2002; Jüni, Altman, and Egger

2001). A similar phenomenon affects systematic reviews and meta-analyses, where many identified citations are often whittled down to a highly selected subset through the successive application of inclusion and exclusion criteria. Therefore, exclusions and attrition must be reported, regardless of their cause, to enable the reader to judge the study's validity and generalizability. It is a standard practice to report, often in the form of an "attrition diagram", how many records were excluded and for what reasons. A well-constructed attrition diagram quickly summarizes each exclusion and the number and percentage of the records affected. The CONSORT statement, a widely adopted set of guidelines for best practices in reporting on clinical trials, encourages the use of attrition diagrams for reporting on exclusions and attrition (Moher, Schulz, and Altman 2001).

It is relatively easy to construct an attrition diagram using a graphics package or a modern word processor. It is harder to ensure the accuracy of the diagram, because every time the data or the criteria change, the analyst has to update the diagram manually. In complex studies with multiple reports, different analyses may have different inclusion and exclusion criteria. Faced by this challenge in our own practice, we created the `attrition` command, which simplifies the maintenance and documentation of the criteria and conditions used to define subsets of the data for a particular analysis and automatically generates a print-ready attrition diagram. The benefit of this approach is that the attrition diagram and the corresponding analysis remain synchronized. This decreases redundancies in the code and eliminates the possibility that exclusion criteria are applied only in one place or missed in manual processing steps.

In this article, we describe the `attrition` command and provide examples of its use.

2 The attrition command

The `attrition` command is a suite of commands that implement exclusion criteria and attrition conditions in a particular analysis. The user provides the name of a binary numeric variable, henceforth *inclusionvar*; a value of 1 indicates that the record is retained in the analysis, and a value of 0 indicates that it is excluded from the analysis. This variable should initially be set to 1 to indicate that the entire dataset should be included before any record is excluded. The variable, along with an `if` or `in` qualifier that should evaluate to true if the record is to be excluded, is passed as an argument to successive calls to the `attrition exclude` command. An optional description string can be used to document this step when eventually included in the attrition diagram. *inclusionvar* can be used in subsequent analyses to limit the analyses to those records that meet all the inclusion criteria. Multiple inclusion variables identifying different subsets of the data are permitted.

To draw the attrition diagram, the user calls the `attrition graph` command with a valid filename as its only argument. This command generates a scalable vector graphics (`.svg`) file (World Wide Web Consortium [W3C] 2011), which can be viewed in any modern web browser. Scalable vector graphics, based on XML (Extensible Markup Language), is a vector image format scalable to any size without loss of quality and created as an open standard. In addition to browsers, many applications, including the

open-source Inkscape package (Kirsanov 2009), can be used to convert `.svg` files into other image formats that might be required for submission to journals. The `attrition tab` and `attrition count` commands can be used to insert boxes containing additional information in the attrition diagram. The `attrition tab` command inserts into the diagram a summary table of included records by one or more variables as if the Stata `tabulate oneway` command were executed with these variables as its argument. The `attrition count` command inserts into the diagram the number of included records (akin to executing the Stata `count` command). The commands can be used as part of a script or interactively through the command line.

2.1 Syntax

Declare a binary variable *inclusionvar* to be tracked as the basis for an attrition diagram:

`attrition set` *inclusionvar* [`, description("`*string*`")`]

Exclude records based on a certain condition specified using an `if` or `in` qualifier:

`attrition exclude` *inclusionvar* [*if*] [*in*] [`, description("`*string*`")`
`noremaining`]

Insert a box into the attrition diagram showing the number of remaining (included) records thus far:

`attrition count` *inclusionvar* [`, description("`*string*`")`]

Insert one or more boxes into the attrition diagram showing the number and percentage of included records stratified by *tabvar*:

`attrition tab` *inclusionvar tabvar*

Prints to the Results window a list of all attrition steps performed thus far:

`attrition list` *inclusionvar*

Graph and save the attrition diagram as an `.svg` file:

`attrition graph` *inclusionvar* `using` *filename* [`, replace` *boxoptions*]

inclusionvar is used to specify the name of an existing binary variable to be used to track which records are included.

filename is used to specify the `.svg` file to which the diagram is written.

2.2 Parameters and options

The commands have the following options:

`description("`*string*`")` provides a way to label each step in the attrition diagram.

`noremaining` hides the number of remaining records in each box.

`replace` overwrites an existing `.svg` file without warning.

boxoptions determine the placement and size of the boxes in the diagram (see table 1).

Table 1. Available *boxoptions*

boxoptions	Description
`width(`#`)`	width of regular (nontabbed) text boxes
`height(`#`)`	height of regular (nontabbed) text boxes
`space(`#`)`	space between text boxes
`fmt_n(%`*fmt*`)`	format for printing numbers
`fmt_pct(%`*fmt*`)`	format for printing percentages

2.3 Implementation

The implementation of `attrition` uses default Stata commands and syntax such as `count` (see [D] **count**), `tabulate oneway` (see [R] **tabulate oneway**), and `replace` (see [D] **generate**) internally to account for steps and generated output text. Details of these executed attrition steps, that is, the text for the attrition diagram, are stored using Stata file characteristics ([P] **char**), which means attrition sequences are saved in the `.dta` file, and are available to generate graphs, even if the original code is not.

Stata's *User's Guide*, [U] **12.8 Characteristics**, refers to characteristics as "an arcane feature of Stata". The feature is not widely known or used, but it allows a programmer to store any text as a characteristic within a dataset as long as it is limited to 13,400 characters for Small Stata and 67,784 characters for Stata/IC, Stata/SE, and Stata/MP. These limits are reasonable because each individual attrition step, which is a line in the printed output or a box in the graph, would typically be limited to at most several hundred characters.

3 Examples

The following code, run using either a do-file or the command line, creates the attrition diagram shown in figure 1.

A binary variable `use_record` is created and initialized to 1. `attrition set` declares `use_record` as the *inclusionvar* for this analysis. Several `attrition exclude` commands are used to successively flag records as excluded based on certain criteria (specified using standard `if` qualifiers). The final sample size is displayed at the end (using an `attrition count` command), and the included records are shown broken down by category (using an `attrition tab` command). The `attrition graph` command then generates the graph shown in figure 1. The size, text spacing, and display formats are all customizable.

```
. webuse lowbirth2
(Applied Logistic Regression, Hosmer & Lemeshow)
. generate use_record = 1
. label define bw 0 "Normal birth weigth" 1 "Low birth weight"
. label val low bw
. attrition set use_record, description("Total births:")
Total births: 112
. attrition exclude use_record if age <18,
> description("Maternal age at birth <18")
(16 real changes made)
. attrition exclude use_record if ht,
> description("Patient has maternal hypertension")
(9 real changes made)
. attrition exclude use_record if ui,
> description("Patient has uterine irritability") noremaining
(18 real changes made)
. attrition count use_record, description("Final sample size:")
  69
. attrition tab use_record low
. attrition graph use_record using "example-birth-weight-output.svg", replace
```

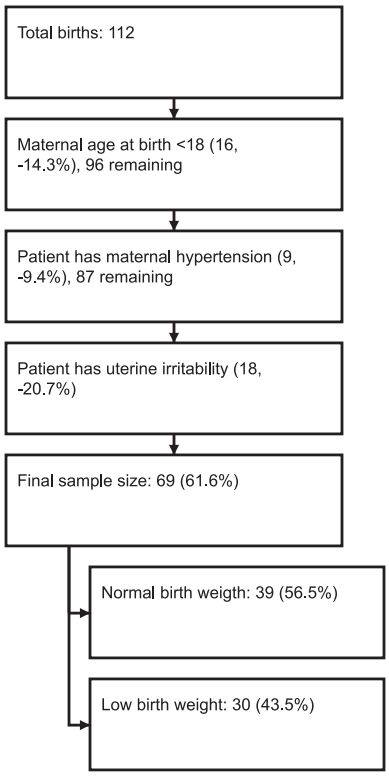

Figure 1. The output diagram of `attrition graph` for the example in the text.

The `attrition list` command generates the following output:

```
. attrition list use_record
Attrition diagram steps for variable use_record.
Total births: 112
Maternal age at birth <18 (16, -14.3%), 96 remaining
Patient has maternal hypertension (9, -9.4%), 87 remaining
Patient has uterine irritability (18, -20.7%)
Final sample size: 69 (61.6%)
  =>Normal birth weigth: 39 (56.5%)
  =>Low birth weight: 30 (43.5%)
```

`use_record` can be used in standard `if` qualifiers to limit all subsequent analyses to included records, for example,

```
logistic low age smoke if use_record
```

Alternatively, only the included records can be kept to simplify the syntax in the analysis:

```
keep if use_record
logistic low age smoke
```

4 Conclusions

In this article, we presented `attrition`, a suite of commands to simplify the maintenance and documentation of implemented exclusion criteria and attrition conditions using standard Stata facilities and to generate an attrition diagram. This workflow keeps the attrition diagram and its documented analysis synchronized automatically, reducing errors that might result from having to recall to update the graph every time the data or the criteria change. The implementation uses Stata file characteristics ([P] **char**) to store all executed attrition steps, which means that attrition sequences are saved in the `.dta` file and are available to generate graphs even if the original code is not. The implementation also allows the user to maintain multiple inclusion variables at the same time, for example, one analysis might be limited to adolescents and another to seniors, while maintaining both attrition sequences simultaneously. A user might also use this to split a large graph into a sequence of smaller, continuing graphs. The `.svg` file can be viewed in any browser and can easily be converted to different image formats using a vector graphics editor such as Inkscape (Kirsanov 2009).

While this command offers a versatile way to build an attrition diagram that can accommodate a variety of study types, it still lacks certain features such as branching criteria (where two or more different sets of criteria are applied to different subsets of the study population). We plan to introduce this feature in a future version.

5 References

Fergusson, D., S. D. Aaron, G. Guyatt, and P. Hébert. 2002. Post-randomisation exclusions: The intention to treat principle and excluding patients from analysis. *British Medical Journal* 325: 652.

Jüni, P., D. G. Altman, and M. Egger. 2001. Assessing the quality of randomised controlled trials. In *Systematic Reviews in Health Care: Meta-Analysis in Context*, 2nd edition, ed. M. Egger, G. Davey Smith, and D. G. Altman, 87–108. London: BMJ Books.

Kirsanov, D. 2009. *Book of Inkscape*. San Francisco: No Starch Press.

Moher, D., K. F. Schulz, and D. G. Altman. 2001. The CONSORT statement: Revised recommendations for improving the quality of reports of parallel-group randomised trials. *Lancet* 357: 1191–1194.

World Wide Web Consortium [W3C]. 2011. Scalable vector graphics (SVG) 1.1 specification. https://www.w3.org/TR/2003/REC-SVG11-20030114/.

About the authors

Christiaan H. Righolt leads the analytics team at the Vaccine and Drug Evaluation Centre. He has extensive experience in quantitative data analysis, both within and outside the realm of epidemiology. He received his doctorate from the Delft University of Technology in the Netherlands, where his thesis focused on the analysis of the nuclear architecture of cancer cells.

He joined the Vaccine and Drug Evaluation Centre in spring 2016 and has worked on several of the tools developed there, including both pharmacoepidemiology-specific and general tools that make data analysis and data interpretation easier in SAS and Stata. His domain-specific work includes research in the areas of human papillomavirus vaccines in relation to human papillomavirus-related diseases, cancer chemoprevention, influenza vaccines, and various other vaccines and their related diseases.

Salaheddin M. Mahmud is a clinician, public health physician, and epidemiologist with interest in drug and vaccine development and evaluation. He is a Canada Research chair in pharmacoepidemiology and vaccine evaluation, the founding director of the Vaccine and Drug Evaluation Centre, the founding director of the Clinical Trials Platform of the Manitoba Canadian Institutes of Health Research SUPPORT Unit, a clinician researcher at Population & Public Health, Manitoba Health, and a member of the Royal Society of Canada's College of New Scholars, Artists, and Scientists. He currently holds an appointment as an associate professor of community health sciences and pharmacy in the Rady College of Medicine, University of Manitoba; is an adjunct professor at the Division of Cancer Epidemiology, McGill University; and serves as the associate editor of *Preventive Medicine* and as a member of several national and international committees and advisory boards related to vaccine policy and development.