# The **tvtools** package for R

David Shilane

July 22, 2012

The **tvtools** software package is designed to aid in the analysis of long–form time–varying (LFTV) data structures. Long–form data allows for each subject in the study to have multiple rows of information. Each row of data corresponds to the subject's status in a specific time interval. Applications such as medical studies may include dozens or hundreds of records for each of tens of thousands of patients. As such, a time–varying data structure is vastly larger than a typical baseline study. Furthermore, the introduction of a time axis adds complexity to the typical descriptive statistics, tables, and figures used to explore the data set. The **tvtools** package provides a variety of methods to aid in the description and visualization of long–form time–varying data.

## 1 Long–form time–varying data

Let's consider the following (artificial) example of an LFTV data set:

| | ID | time1 | time2 | age | drug | death |
|---|---|---|---|---|---|---|
| Row 1 | 1 | 0 | 5 | 65 | 1 | 0 |
| Row 2 | 1 | 5 | 6 | 65 | 1 | 1 |
| Row 3 | 2 | 0 | 3 | 60 | 1 | 0 |
| Row 4 | 2 | 3 | 10 | 60 | 0 | 0 |
| Row 5 | 2 | 10 | 12 | 60 | 1 | 0 |
| Row 6 | 3 | 0 | 8 | 70 | 0 | 0 |
| Row 7 | 3 | 8 | 9 | 70 | 1 | 1 |
| Row 8 | 4 | 0 | 2 | 85 | 0 | 1 |
| Row 9 | 5 | 0 | 6 | 55 | 1 | 0 |
| Row 10 | 5 | 6 | 15 | 55 | 0 | 0 |

Table 1: Example of a LFTV data set.

Each row includes the following information:

- **An ID field**: This ensures that a subject with multiple rows will have all of its information linked together by the identification variable.
- **Time intervals**: Each row represents the subject's status in a specific interval of time. The left and right endpoints of this interval are given by the variables **time1** and **time2**, respectively. Other names may be substituted. In general, the time intervals for a given

subject must be mutually exclusive. In most cases, they will also be collectively exhaustive. The first time interval for each subject typically begins at 0, and the last interval typically ends at the maximum follow–up time for the subject. We will assume that the subject's status changes instantaneously at the left end–point of each interval from the previous to the current status. Moreover, this status will remain fixed for the duration of the time interval.

- **Baseline covariates**: Some variables are measured only a single time (e.g. at baseline) or are otherwise unchanging. The LFTV data structure repeats these variables with the same value in each row.

- **Time–varying data**: These variables may change within each time interval. Depending upon the context, some variables have limits on how they may change with time. For instance, in survival analyses, a measure of mortality may change from 0 to 1 but not back. Other outcomes may repeatedly alternate between states. In medical studies, the time of certain events (e.g. myocardial infarction) may be recorded. These variables will take the value 1 at the time of the event and then return to 0 in the next time interval.

The example in Table 1 depicts a study examining the effect of age and a drug on survival. In this case, the time units are in months. The first patient entered the study at age 65, was treated with the drug, and survived for 5 months. The second patient entered the study at age 60, received 3 months of drug treatment, spent months 3–10 off of the drug, reinitiated the treatment from months 10–12, and survived at least until the end of the 12 month follow–up.

The stories of each patient are easy enough to understand if you read through each row. However, some very simple questions are more difficult to answer with the basic tools of descriptive analysis. For instance, we might ask:

- What was the average drug exposure for each patient? This would be the number of months on treatment divided by the overall number of months each patient was observed. However, since patients are followed over multiple rows, the follow–up time must be identified, and the treatment variable must be weighted in each row by the length of the time interval.

- What was the death rate for patients on and off of the drug? The number of deaths in each category may be counted up within the subset of rows on and off the drug treatment. Then these counts must be weighed against the overall time of exposure. That is, instead of simply saying that a certain percentage survived past 6 months, we must provide an answer in terms of survival per person per month.

- How soon do patients tend to die after initiating the drug treatment? We must first identify the time that each patient started the drug (if ever). Then we may compare this date to the time of death.

Each of these questions has a temporal nature in addition to the usual statistical questions of rates or averages. The usual tools of descriptive statistics require some augmentation to arrive at the proper answer. The **tvtools** package seeks to provide effective methods for solving these problems. The goals of the **tvtools** package are:

1. **Compute crude rates** of exposure and event occurrences. This may include:

- **Cross-sectional rates:** This can be the percentage of the population with a certain condition at a specific time. Note that time intervals beyond some patients' length of follow–up will be accounted for.
- **Event rates relative to exposure:** This will compute factors such as the death rate per person year of follow-up.
- **Event rates relative to exposure and split by treatment category:** This will compute factors such as the death rate per person year on and off of drug treatment.

**Implemented in cruderates.R**

2. Compute time to events. **Implemented in firstevent.R**
3. Compute overall time of follow–up for each patient. **Implemented in followuptime.R**
4. Compute rate of exposure over a specific time frame. (E.g. the percentage of days exposed to drug treatment in the first year.) **Implemented in exposure.R**
5. Create treatment exposure graphics:
   - Depict drug exposure timelines for multiple simultaneous treatments.
   - Display timelines of events

   **Implemented in timeplot.R**
6. Create a cross-sectional data set at a specific time. For baseline data sets, include the time to specified events. **Implemented in create.baseline.R.**
7. Account for the degree of missingness in a variable at specific time points. **Implemented in missingness.R.**
8. Provide visualizations to track the percentage of missingness in a variable across time. **Implemented as missingness.plot() in missingness.R.**

Figure 1: One patient's medical history.

## 2 Functions

# Computing Crude Rates in Time-Varying Data

### Description

**cruderates** is used to compute means and percentages for time-varying variables. These crude rates are in units of both people and time (e.g. an average of 10 myocardial infarctions per person-year of followup). The rates may be split across treatment categories or computed separately in different eras of follow-up.

### Usage

cruderates(dat, tx.name=NA, outcome.names, time.names=c("t1", "t2"), time.units="day", cut.points=NA, result.units="year", era.digits=0)

### Arugments

dat: a data frame

tx.name: A character object containing the name of the treatment variable. The variable must be in names(dat). When specified, rates are computed separately for each value of the treatment variable (e.g. treatment cases and control cases). Overall rates for the entire data set are always computed whether tx.name is specified or not.

outcome.names: A character vector containing the names of the variables to be summarized. All values in outcome.names must be variable names in names(dat).

time.names: A character vector of length 2 indicating the beginning and end of each time interval. Both values of time.names must be in names(dat). Typically the beginning of the interval is specified first, but the software will automatically correct mis-orderings.

time.units: A character value indicating the data's unit of time. Possible values include "day", "month", or "year". Unless otherwise specified, the data will be as-

sumed to be recorded in days of follow-up.

cut.points: A numeric vector indicating separate eras in which to compute crude rates. For instance, one may wish to compute the mortality rate in 0-6 months, 6-12 months, and after 12 months. By default, the beginning and end of follow-up are always included, so only the cut points need to be specified. In the previous example, this would be accomplished by cut.points=c(6, 12) if time.units="month". If time.units="day", then this would be represented with cut.points=c(0.5*365, 365).

result.units: A character value indicating the time component of the crude rates calculation. Possible values include "day", "month", or "year". Unless otherwise specified, the rates will be represented in terms of person-years.

era.digits: A numeric value indicating how many digits to round the cut.points to in the display of the crude rates.