Farewell's Linear Increments Model for Missing Data: The FLIM Package

by Rune Hoff, Jon Michael Gran and Daniel Farewell

Abstract Missing data is common in longitudinal studies. We present a package for Farewell's Linear Increments Model for Missing Data (the **FLIM** package), which can be used to fit linear models for observed increments of longitudinal processes and impute missing data. The method is valid for data with regular observation patterns. The end result is a list of fitted models and a hypothetical complete dataset corresponding to the data we might have observed had individuals not been missing. The **FLIM** package may also be applied to longitudinal studies for causal analysis, by considering counterfactual data as missing data - for instance to compare the effect of different treatments when only data from observational studies are available. The aim of this article is to give an introduction to the **FLIM** package and to demonstrate how the package can be applied.

Introduction

Longitudinal data consist of repeated measurements recorded for a group of individuals over a given time period. The resultant datasets are often incomplete, i.e. missing responses for some individuals. Incomplete datasets are more troublesome to analyse, since one has to assess the mechanism behind the missing data. If a response measure itself is linked to the probability of being missing, a standard analysis of the observed data may be severely biased.

Data from repeated observations have a natural time order. Standard likelihood-based models for quantitative analysis usually ignore time order and treat longitudinal data as cluster data (Hogan et al., 2004), giving little regard to where missingness takes place or to how the response processes evolve. Farewell (2006) and Diggle et al. (2007) introduced the linear increments model as a tool for dealing with missing data due to drop-out in a dynamic manner, that is, explicitly considering the time order of the responses and where drop-out occurred. The key concept in the linear increments model is the increments, which represent changes over time, and hence are representative of evolving response processes (Aalen and Gunnes, 2010).

As with any method that considers missing data, the linear increments model is not exempt from untestable assumptions. In the linear increments model, the discrete time-independent censoring (DTIC) assumption is adopted. This assumption states that the local characteristics of the response are unchanged by additional information about who is censored, or knowledge of who will, or will not, be censored at the next time point (Diggle et al., 2007). The DTIC assumption is similar to the sequential version of missing at random assumption (Hogan et al., 2004), and to independent censoring in survival analysis.

The **FLIM** package implements an autoregressive multivariate version of the linear increments model described by Aalen and Gunnes (2010). The end result is a hypothetical complete dataset that might have been observed had individuals not been missing, referred to as a hypothetical complete dataset, which can be used for further analysis. Because of the nature of the method, it is required that the longitudinal data exhibits a regular observation pattern with the same potential observation times for all individuals. The **FLIM** package may also be applied to longitudinal studies for causal analysis, by considering the counterfactuals as missing data. For instance, we may artificially censor individuals after they initiate treatment and apply the linear increments model to estimate counterfactual responses had treatment not been initiated.

Early work on the linear increments model focused on handling monotone missingness (i.e. when individuals are missing and do not return to the study; typically referred to as drop-out). The approach of Aalen and Gunnes (2010), which we adopt, includes both monotone and nonmonotone missingness (i.e. when individuals are missing for a certain amount of time, but later return to the study), and the required assumptions for both types of missingness are discussed.

The linear increments model

The linear increments model assumes a hypothetical complete dataset. The focus lies in the response that would have been observed if, contrary to fact, individuals did not drop out. Thus, the linear increments model makes explicit the possibility that dropping out can influence the response measurements, rather than just lead to missing data.

The model is, as the name implies, based on linear models for the increments, which represent changes over time and thus capture the evolving response process (Aalen and Gunnes, 2010). At every time point, a linear model is fitted for each longitudinal response by regressing the observed increments onto lagged values of the response variables and possibly other covariates. Afterwards, missing data are imputed by estimating increments for the unobserved individuals and then adding these to the lagged values of their response variables.

To fix some notation, we start by describing the hypothetical complete dataset. Let $\tilde{Y}(t)$ be an $n \times m$ matrix of multivariate individual responses defined for a set of times $t \in \{0, ..., k\}$, with $\tilde{Y}(0) = y_0$ being a matrix of fixed starting values for the processes. Further we write $X(t)$ for an $n \times p$ matrix of multivariate individual covariates that can contain both baseline and time dependent variables – note that any time dependent covariate is assumed to take on the same value as the last observed value on missing observations. Aalen and Gunnes (2010) define the increment of a response measure as $\Delta\tilde{Y}(t) = \tilde{Y}(t) - \tilde{Y}(t-1)$, and we assume for each t that $\Delta\tilde{Y}(t)$ satisfies the model

$$\Delta\tilde{Y}(t) = X(t-1)\alpha(t) + \tilde{Y}(t-1)\beta(t) + \tilde{\epsilon}(t). \tag{1}$$

Here $\beta(t)$ is an $m \times m$ matrix of parameters, $\alpha(t)$ is an $p \times m$ matrix of parameters and $\tilde{\epsilon}(t)$ is an $n \times m$ error matrix. The error is defined as a zero mean martingale (Diggle et al., 2007). It follows that

$$\mathbb{E}(\Delta\tilde{Y}(t)|\mathcal{F}_{t-1}) = X(t-1)\alpha(t) + \tilde{Y}(t-1)\beta(t),$$

where $\mathcal{F}_t$ is the history of responses, and covariates up to and including time t.

To estimate the parameters in (1), a nonparametric model is assumed over time, so there is no connection between $\alpha(t_1)$ and $\alpha(t_2)$ for different times t_1 and t_2 and neither for $\beta(t_1)$ and $\beta(t_2)$. The parameter matrices $\alpha(t)$ and $\beta(t)$ can be estimated in an unbiased manner by ordinary least squares from the observed increments (Aalen and Gunnes, 2010).

In order to simplify the derivations, we shall for the rest of this section assume that there only are responses and no covariates. To write out the estimator for $\beta(t)$, we define $R(t)$ and $R_0(t)$ as $n \times n$ diagonal matrices with response indicators on the diagonal. The ith diagonal element of $R(t)$ equals 1 when the complete set of response increments $\Delta\tilde{Y}_i(t)$ is observed for individual i, and $R_0(t)$ is defined similarly for the response measures. We assume that observation of an increment $\Delta\tilde{Y}_i(t)$ implies that both $\tilde{Y}_i(t)$ and $\tilde{Y}_i(t-1)$ were observed. Define $Y(t)$ to be the observed data, so the increments can be written as $\Delta Y(t) = Y(t) - Y(t-1)$ when they are well defined. Thus it is not necessarily the case that the observed values $Y(t)$ equals the sum $\sum \Delta Y(t)$. If we now write $U = R(t)Y(t-1)$, the least squares estimate of $\beta(t)$ is:

$$\hat{\beta}(t) = (U^T U)^{-1} U^T \Delta Y(t). \tag{2}$$

Once linear models for the observed increments are fitted, estimates of the increments for missing individuals may be calculated and data can be imputed. The idea is to use real responses where they are available and then to substitute missing increments by iteratively updated estimates. The imputation scheme for the predicted hypothetical complete data is as following:

$$\begin{aligned}
\tilde{Y}^{est}(0) &= y_0 \\
\Delta\tilde{Y}^{est}(t) &= (1 - R_0(t))\tilde{Y}^{est}(t-1)\hat{\beta}(t) + R_0(t)(Y(t) - \tilde{Y}^{est}(t-1)),\ t = 1,...,k, \\
\tilde{Y}^{est}(t) &= \tilde{Y}^{est}(t-1) + \Delta\tilde{Y}^{est}(t),\ t = 1,...,k.
\end{aligned}$$

When data is observed, the value $\tilde{Y}^{est}(t)$ just equals $Y(t)$. When there is missing data, the increments are predicted according to the model.

When applying the linear increments model, some issues might need to be resolved for the imputation to work. To obtain reasonable estimates, a sufficient number of observed increments must be available at all times. Sufficient number is a vague concept, but towards the end of a study there are typically less individuals left. Therefore, it might sometimes be advisable to stop imputation before the last observation times. Also, if the matrix $U^T U$ in (2) is singular at any time, such that it cannot be inverted, the routine is halted. At this point, one option is to do a Ridge regression to find β-estimates.

The DTIC assumption requirement for the data is discussed below – if the required assumptions hold, hypothetical complete datasets constructed using the above procedures will have the correct mean structure. However, they will not have the correct random error variation (Aalen and Gunnes, 2010). To deal with this, bootstrap samples of hypothetical complete datasets can be produced. This should not be too time-consuming, as iterations are not needed to find estimates. The bootstrap is done so that the same number of individuals as in the original data are sampled with replacement from the study population. If an individual is sampled, his or her entire observation matrix should be included. Missing data in a bootstrap sample are then imputed in the same manner as the original dataset, and inference can be based on these bootstrap samples. In the package, we have included the possibility to calculate empirical bootstrap estimates of the standard errors for the hypothetical mean responses, and also an option to draw point wise 95% confidence bands around the hypothetical mean response curves.

To perform model check for the increments models, one can use the standard diagnostics plots for linear regression fits. These plots should reveal any serious misspecification of the model, and users should check for the usual signs of model misspecification in linear models – these are briefly explained when demonstrating the package later. We have included in the package an option to easily view the diagnostics plots that are produced for `"lm"` objects in R on the different time points where linear models were fitted.

The discrete time-independent censoring assumption

The DTIC assumption is covered with greater detail in Gunnes et al. (2009). Despite the term censoring, the DTIC is actually an assumption about missingness. The terminology comes from the analogy of independent censoring in survival analysis.

To write out the assumption, let $\mathcal{R}(t)$ be the past history of $R(t)$, i.e the history of the censoring process up to time $t-1$. The censoring is considered to be independent of the hypothetical response process $\tilde{Y}$ if, and only if,

$$E(\Delta\tilde{Y}(t)|\mathcal{F}_{t-1},\mathcal{R}(t)) = E(\Delta\tilde{Y}(t)|\mathcal{F}_{t-1}), \forall\, t. \tag{3}$$

Independent censoring says that the local characteristics of $\tilde{Y}$ are unchanged by additional information about the censoring process. This assumption ensures that the observed increments remain representative of the original study sample (Diggle et al., 2007). The DTIC assumption guarantees that the observed data will satisfy the model specified in (1) (Aalen and Gunnes, 2010).

Monotone and nonmonotone missingness

Early works on the linear increments model concentrated on censored or monotone missing data. Although the imputation technique may be valid when there are nonmonotone missing data, assumptions can prove harder to justify, and it might therefore be necessary to remove data from individuals with nonmonotone missingness. In many cases, if data are missing at only a few time points it would be foolish to ignore subsequent data. For instance, if only one measurement is missing for reasons unrelated to the evolution of the response process, then later measurements for the same individual should not be ignored. On the other hand, individuals who are missing from the study for an extended period and then return might not be representative of the population of the missing ones. The problem is to decide whether the data from these individuals should be used, which goes beyond the DTIC assumption (Aalen and Gunnes, 2010).

To decide whether monotone or nonmonotone missing data can be used, one must determine whether the probabilities of going missing and returning to the study are related to the progress of the complete response $\tilde{Y}(t)$. There are four possible situations,

1. Going missing "is unrelated" to the previous response progress. Returning "does not depend" on the progress while missing.
2. Going missing "is related" to the previous response progress. Returning "does not depend" on the progress while missing.
3. Going missing "is unrelated" to the previous response progress. Returning "does depend" on the progress while missing.
4. Going missing "is related" to the previous response progress. Returning "does depend" on the progress while missing.

In order to consistently use nonmonotone missing data, the probability of returning to the study should not depend on the response progress while missing, as is the case in situations 1 and 2 above. In situations where the probability of returning does depend on the response process while missing, data from observation times after an individual returns may need to be removed from the analysis. For all four situations above, monotone missing data can be used, since no considerations to any developments during absence have to be made. For a more elaborate discussion on the matter, see Aalen and Gunnes (2010).

Causal analysis

Methods for dealing with missing data have much in common with methods for estimating causal effects (Aalen and Gunnes, 2010). A typical problem in longitudinal studies is the comparison of treatment effects when treatment is confounded with the condition of the individual rather than being randomised. For instance, if treatment is initiated with increasing probability as individuals get sicker, a naive analysis could conclude that individuals receiving treatment are worse off than if left untreated. Here the linear increments model can be adopted to estimate the counterfactual condition progress had treatment not been initiated, which could shed light on the treatment effectiveness. To see how this can be achieved, note that, as reported by Hernán et al. (2006), treatment initiation or change in treatment may be seen as artificial censoring.

To demonstrate how the linear increments model can be used to assess causal effects, for example, assume a situation where treatment A and treatment B are to be compared in their effectiveness of improving the condition of the patients. Rather than randomising treatment, the decision regarding type of treatment is made based on the patients' conditions, typically quantified by some measured response variable. Patients are given either treatment A or treatment B, and those receiving treatment A are moved to treatment B if their condition worsens. Depending on the purpose of the analysis, we may explicitly censor response values for individuals who changed from treatment A to treatment B after their last observation under treatment A. The data would be reconstructed as if these patients never initiated treatment B, but rather continued treatment A and were "missing" from follow-up for the remainder of the study.

It is possible to use **FLIM** to impute counterfactual response developments as described above. To do this, the dataset should include a factor variable that indicates that an intervention has taken place, for instance some form of treatment. This variable should be coded so that the value is zero whenever a patient is not intervened on, and 1 from the time point when intervention first took place. Intervention has to be of such nature that once initiated, the patient continuous to undergo this intervention for the remainder of the study period or until being censored or having an event. An example is shown on a simulated dataset later in the article.

The FLIM package

Applications of the linear increments model in the **FLIM** package are done with the core function `flim`:

```
flim(formula, data, id, obstime, t.values = NULL, method = "locf", lambda = NULL,
       art.cens=NULL)
```

To use the function, the following *mandatory arguments* must be entered. Note that `id` and `obstime` should be entered with quotation marks.

- `formula`: an R formula on the form `response ~ predictors`. The package fits models for the increments, so that `formula = Y ~ Y + X` specifies the model
  $E[Y(t+1) - Y(t)] = \beta_0(t) + \beta_1(t)Y(t) + \beta_2(t)X(t)$
  For several responses and the same set of predictors, `cbind(Y1,Y2) ~ predictors` can be used. For full flexibility, a list of formulae can be supplied.
- `data`: a data frame with the longitudinal dataset. See separate section later for specification requirements.
- `id`: name of the variable in `data` with unique ids for each subject.
- `obstime`: name of the variable with observation times.

In addition, there are some *optional arguments* that should be specified accordingly when needed.

- `t.values`: a vector with time points at which models should be fitted and missing data imputed. Models are fitted from the first time point all the way through to the penultimate time point. While data are imputed from the second time point up to the last one. If nothing is specified, this argument will be set to the unique time points contained in the data.
- `method`: method for filling in response values between observations if there is nonmonotone missingness. The options are "locf", "approx", "recursive" and "recursiveX". Default is "locf". See details in a separate section about this argument.
- `lambda`: ridge parameter for doing ridge regression in the linear fits. Default is OLS.
- `art.cens`: used to specify a 0-1 factor variable that will be used for artificial censoring. Artificial censoring is performed to response values for all subjects after the first switch from 0 to 1. Missing values will then be imputed as if subjects never made the switch. This is intended for advanced users and does not come with readily available tools for investigating results. The original dataset with added columns for counterfactual response values is stored in the fitted object.

Below is an example of how arguments can be entered. Imagine a dataset called `mydata` consisting of a cohort of HIV patients similar to e.g Aalen et al. (2012). The patients are observed regularly, where they have among other variables, measured cd4 and rna levels. Suppose ids and observation times are stored in the columns `pID` and `Obs`, respectively, and we wish to impute missing values for the two responses, `cd4` and `rna`. We also want to include the covariate `age` as a predictor. Assume our goal is to fit models and impute data for times `t` $\in \{1, 2, 3, ..., 20\}$ – individuals may be observed on any arbitrary set of times contained within `t` as long as they are fully observed on their first observation, that is, all responses and covariates are registered. Any potential nonmonotone missing data are to be filled in by last observation carried forward (LOCF). To achieve this, we enter the following:

```
flimobject <- flim(cbind(cd4, rna) ~ cd4 + rna + age,
                   mydata, "pID", "Obs", t.values = 1:20)
```

This example would be equivalent to specifying these two models:

$$\begin{aligned} E[cd4(t+1) - cd4(t)] &= \beta_0(t) + \beta_1(t)cd4(t) + \beta_2(t)rna(t) + \beta_3(t)age \\ E[rna(t+1) - rna(t)] &= \alpha_0(t) + \alpha_1(t)cd4(t) + \alpha_2(t)rna(t) + \alpha_3(t)age \end{aligned}$$

A short stepwise review of the programme flow is as follows. First data are sorted by id and increasing time. Then a check for whether all individuals have a complete set of responses on their first observation is performed. After this, the dataset is prepared for imputation so that each individual has rows corresponding to the same set of observation times. If there is nonmonotone missingness, this is handled according to the `method` argument, which is described in greater detail later in the article. Subsequently, increments are calculated, and separate regressions performed at each time point.

To combine the reconstruction driven approach where missing data are imputed, and the task of fitting models for the increments, the linear regressions are performed successively by increasing observation time. This means that missing data can routinely be inserted based on previous imputed data and the model for increments at the current time.

The end result from applying `flim` is an object of class `"flim"` in R – a list containing among others the value `dataset` with the reconstructed data (hypothetical complete dataset) and `fit`, which contains the fitted models. Methods and functions for the class `"flim"` are as follows:

- `print(x)`
- `summary(x)`
- `plot(x,...)`, see separate section on how to plot a `"flim"` object for additional arguments.
- `flimMean(x,response,grouping=NULL)`, calculates mean responses.
- `flimList(x)`, to assess model fits.
- `flimboot(x,R,counter = F)`, bootstraps a flim object.

The arguments are

- `x`: an object of class `"flim"`.
- `response`: name of a response variable.
- `grouping`: optional factor variable used in the model.
- `R`: number of bootstrap samples.
- `counter`: logical. If TRUE displays a bootstrap sample counter. Works by default on Mac and Linux platforms.

The functions `flimList` and `flimboot` take a fitted `"flim"` object as main argument and create objects of class corresponding to the function name. Examples are shown in the application section later. `flimList` is used to assess the model fits and has three usages

- `print(flimList(flimobject))` prints the predictor coefficients for all increment models on every included time point.
- `summary(flimList(flimobject))` gives estimated standard errors, t-values and p-values for all predictor coefficients in the linear models at every time point.
- `plot(flimList(flimobject,response))` plots model diagnostics in a 2x2 grid for the linear model for the specified `response`. Plots are shown for the first time point, then the second and so on when pressing `<ENTER>`.

Bootstrapping a flim object can be done with `flimboot`, which creates a `"flimboot"` object. The class has two working methods. Suppose `fbo` is a `"flimboot` object, then

- flimSD(fbo,response,grouping = NULL) calculates empirical standard errors of the hypothetical bootstrap mean responses. The argument response chooses the response variable, and grouping is an optional factor variable.
- plot(fbo,response,grouping=NULL,...) plots the hypothetical reconstructed mean response from the original dataset together with bootstrapped point wise 95% confidence bands. Takes the same arguments as flimSD and in addition any argument that can be passed to plot. By default the hypothetical means are fully drawn and the confidence bands stipulated.

The dataset

A long format longitudinal dataset, in which each observation is stored as a separate row, is required. Further, the dataset must contain a column for id, which uniquely identifies every individual in the study, and a column for the observation times. For each individual, the data should only include rows corresponding to observation times where at least one of the (potentially many) response variables is observed, missing data are denoted NA. For the iterative imputation to work, all individuals need a complete set of responses at their first observation time. The panss dataset (Diggle, 1998) included in the package is an example of how the data should look . If data are in wide format, one option is to transform them into long format with the reshape function in R.

The method argument

Specifying this argument in the flim function determines how missing data before drop-out should be treated; after drop-out, the linear increments model is used regardless of choice of method. There are four options, "locf" (default), "approx" "recursive" and "recursiveX".

For the option "locf", missing values when there are available observed values on a later time point, are imputed by inserting the last observed value. To implement LOCF, we use the function na.locf in the **zoo** package (Zeileis and Grothendieck, 2005).

The second option is "approx", or the approximation method, which imputes the missing data between observations by linear approximation, calculated with na.approx in **zoo**. Conceptually this approach may be more appealing than LOCF. The approximation method performs a linear interpolation to impute missing data between two observed values. The pitfall of this approach is that it may go against our prime directive: to consider specifically the time order of the measurements, and to use information contained in the response histories available at the time when data is missing, to impute data. Indeed, the approximation method violates this by looking ahead at future response values, which is in some sense cheating.

The third option is "recursive", which uses the linear increments model to fill in data for missing individuals throughout the entire dataset – regardless of whether they are observed again on a later time or not.

The final option is "recursiveX", which is similar to "recursive". Unlike "recursive", when fitting models, this method utilises data from observations just as patients return to the study, where there are no observed increments since the values are missing until the patients return. This is achieved by using estimated increments based on previously imputed data.

Plotting a flim object

The plot function can be used to plot the hypothetical mean responses of the imputed data, the mean responses of the observed data and a spaghetti plot of all individual trajectories. The default method is to plot mean responses where both hypothetical and observed curves are shown.

```
plot(x, response, grouping = NULL, ylim = NULL, col = NULL, naive = T,
 lty = 1:2, ptype = "mean", ylab = "Response", xlab = "Times", ...)
```

Arguments for plotting a flim object are:

- x: the "flim" object.
- response: name of a longitudinal response.
- grouping: optional group/factor variable used in the model.
- ptype: decides plot type, default is "mean" for mean responses, while ptype="spa" will give spaghetti type plot.
- naive: logical. If TRUE the plot will contain the observed means as well.

The rest of the arguments are standard arguments to `plot` that have been given default values – these may be altered by the user. Note that `lty` is a vector of length two and specifies the linetype of the observed mean response and hypothetical mean response respectively. An example of plotting a `"flim"` object is shown in the application section.

Bootstrapping with FLIM

Once a `"flim"` object is created, it can be bootstrapped to create a list of `"flim"` objects, each using a new dataset that has been resampled from the original data. The resampling is done with replacements on the id numbers so that every time an individual is selected to the bootstrap data, their full observation cluster is included. Bootstrapping in the linear increments model allows us to easily calculate standard errors for the hypothetical mean response.

The bootstrap function is called `bootflim`. Because `flim` can handle a lot of different model specifications and imputes data iteratively, the routine may be slower than a script working for one specific model. Nevertheless, because no iterations are required to reach the estimates in the models, time is not a major issue – with our example data `panss`, 100 samples were fitted in around 10 seconds on a laptop.

After a `"flim"` object have been bootstrapped, the hypothetical mean using imputed data in the original dataset can easily be plotted with confidence bands as shown in the application section – these confidence bands are point-wise and 95%.

Application

In this section, we will analyse data from an investigation of the effect on PANSS score of different schizophrenia treatments. A longitudinal study was conducted to compare three treatment regimens, haloperidol, placebo and risperidone among patients with schizophrenia. Patients were randomised to treatments at time 0, and followed for 8 weeks where they had PANSS score evaluated. PANSS is a score measure that quantifies the severity of schizophrenic symptoms, and the goal of the study was to compare the effectiveness of treatments in improving (reducing) the mean PANSS score. The data consist of 685 observations among 150 patients, and were extracted from a larger, confidential dataset from a randomised clinical trial. An analysis of the complete dataset can be found in Diggle (1998).

```
> install.packages("FLIM")
> library(FLIM)
> data(panss)
> head(panss, 8)
  treat time   Y id
1     1    0  91  1
2     2    0  72  2
3     1    0 108  3
4     1    1 110  3
5     1    0 106  4
6     1    1  93  4
7     1    0  77  5
8     1    1  80  5
```

To fit the linear models for increments and reconstruct the data, one should use the main function `flim`. Here a model where the PANSS score increments are regressed onto the score value and the treatment group is fitted, which corresponds to entering `Y∼Y+factor(treat)` in the function. The remaining mandatory arguments are filled in as needed, and the extra arguments are left unspecified.

```
 panss.flim <- flim(Y~Y+factor(treat), data=panss, id="id", obstime="time")
```

The stored object `panss.flim` is now of class type `"flim"`; some additional information about the object, over the standard `print`, can be had by using `summary` on the object. If users want direct access to the reconstructed data, these are stored in the value `dataset`. The linear models fitted on each time point are stored as a list in the value `fit`.

Instead of directly accessing the data and the models, users may want to take advantage of the built-in functions available to analyse the results. We start by calculating the mean response with `flimMean`:

```
> flimMean(panss.flim, "Y")
  hypothetical observed
```

```
0     92.02000 92.02000
1     87.39672 87.44595
2     86.01734 83.07874
4     85.78134 80.25926
6     87.88868 77.42857
8     88.52504 76.00000
```

This is displaying the mean response of the variable Y for the observed data and the hypothetical reconstructed data.

In order to perform separate calculations for the different treatment groups, the grouping argument must be specified. The first three columns in the print below show the mean responses for the hypothetical reconstructed data, while the last three are for the observed data.

```
> flimMean(panss.flim,"Y", grouping="treat")
         1         2        3    1 obs    2 obs    3 obs
0 93.40000  91.40000 91.26000 93.40000 91.40000 91.26000
1 87.84636  93.54380 80.80000 87.87755 93.79592 80.80000
2 86.60223  92.80858 78.64123 84.47727 88.42105 77.20000
4 87.91717  95.21459 74.21225 83.92500 86.30000 71.63158
6 85.80121 104.10217 73.76264 75.67857 91.52174 69.09091
8 83.48531 103.63105 78.45876 73.72000 86.87500 71.66667
```

The mean responses can be plotted with the plot function – we specify the grouping argument so that separate curves are drawn for each treatment group:

```
> plot(panss.flim,"Y", grouping="treat", col=c("green","blue","brown"))
Mean response plot created for variable: Y
Full drawn: observed mean response.
Stipulated: flim hypothetical mean response.

Response variable separated by: treat which has 3 levels.
Group: 1 has color green. Group: 2 has color blue. Group: 3 has color brown.
```

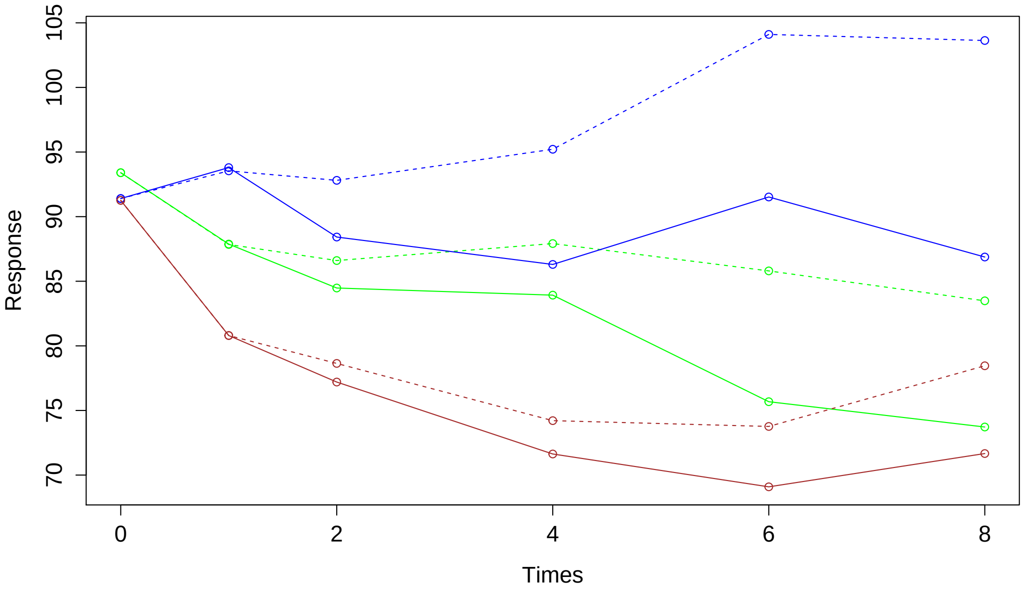

Figure 1: Observed (solid line) and hypothetical (dotted line) mean responses for haloperidol (green), placebo (blue) and resperidone (brown) treatment regimens among patients with schizophrenia.

The different treatments are 1: haloperidol, 2: placebo and 3: risperidone. Legends are not auto generated by the function and must be added by the user if needed. Observed and hypothetical mean responses are shown in Figure 1. Looking at the figure, we find that the haloperidol and risperidone treatment groups show better progress than the placebo group. Moreover, the risperidone group shows fewer symptoms than the haloperidol group. But the most striking feature is that among all treatment groups, there are clear differences between the observed and the hypothetical curves. When considering the observed mean responses, one might infer that PANSS scores improve (decrease) over time regardless of treatment group. Indeed, even the placebo group shows a decreasing observed mean response. On the other hand, when the hypothetical mean responses are considered, the linear increments model suggests that this is due to informative drop-out, and therefore the observed data are underestimating the PANSS score. An increase in symptoms is even suggested in the placebo group, while the haloperidol and risperidone groups show a reduced effect (less decline) when comparing hypothetical to observed mean responses.

By default, the plot function displays the mean response – to show individual trajectories, use `ptype="spa"`. The spaghetti plot is shown in Figure 2, in which solid lines are used before drop out, and the dotted lines are used thereafter, when values are imputed:

```
plot(panss.flim,"Y", grouping="treat", col=c("green","blue","brown"), ptype="spa")
```

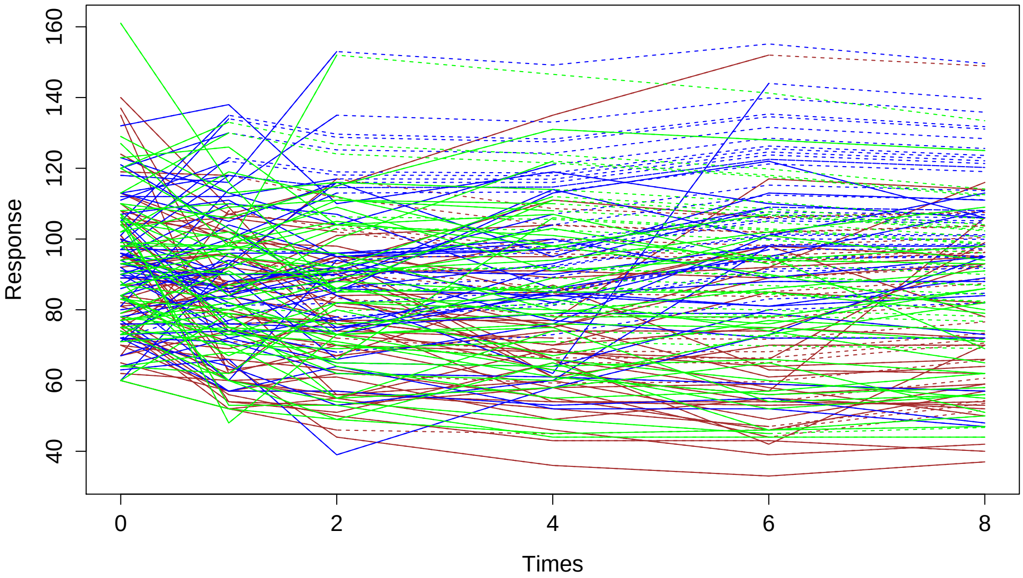

Figure 2: Individual responses for haloperidol, placebo and resperidone treatment regimens among patients with schizophrenia. Dotted lines are used after drop-out.

Assessing model fit

The fitted linear models for the increments are contained in the `flim` object as a list in the value `fit`. Users may perform standard model checking by directly accessing these. However, to view summary of the coefficients and running model diagnostics on the different time points in a more compact and convenient way, the function `flimList` is useful:

```
> flimList(panss.flim)
Call:
  Model: Y.inc ~ Y + factor(treat)
   Data: panss
  Times: 0 1 2 4 6

 Coefficients:
  (Intercept)      Y factor(treat)2 factor(treat)3
0      28.369 -0.363          6.971         -5.684
1       8.557 -0.112          1.145         -1.701
2      10.264 -0.103          1.732         -6.567
4       2.650 -0.054         11.399          0.923
6       6.199 -0.099          3.661          5.817
```

These coefficients represents the effects of responses and covariates on the expected change in the responses at the different time points.

The summary function will display the estimated coefficients, standard errors, t-values and p-values:

```
> summary(flimList(panss.flim))
Call:
  Model: Y.inc ~ Y + factor(treat)
   Data: panss
  Times: 0 1 2 4 6

 Coefficients:

 (Intercept)
   Estimate Std. Error   t value     Pr(>|t|)
0 28.369363   6.766822 4.1924203 4.796753e-05
1  8.557158   6.069304 1.4099075 1.610906e-01
2 10.263738   5.880129 1.7454954 8.385207e-02
4  2.650326   6.697694 0.3957073 6.933743e-01
6  6.199175   5.389000 1.1503387 2.542846e-01

 Y
     Estimate Std. Error    t value     Pr(>|t|)
0 -0.36320130 0.06831664 -5.3164396 3.948642e-07
1 -0.11157312 0.06689537 -1.6678752 9.788447e-02
2 -0.10333217 0.06732753 -1.5347686 1.278774e-01
4 -0.05421329 0.07943341 -0.6824999 4.968945e-01
6 -0.09924193 0.06558725 -1.5131284 1.351699e-01

 factor(treat)2
   Estimate Std. Error   t value    Pr(>|t|)
0  6.971035   3.174670 2.1958296 0.029704587
1  1.144598   3.002800 0.3811768 0.703729668
2  1.732388   3.049484 0.5680922 0.571196889
4 11.399145   4.024242 2.8326193 0.005839657
6  3.661010   3.599667 1.0170413 0.312963201

 factor(treat)3
    Estimate Std. Error    t value   Pr(>|t|)
0 -5.6836117   3.160298 -1.7984418 0.07420171
1 -1.7008236   2.898236 -0.5868478 0.55838151
2 -6.5665452   2.891971 -2.2706126 0.02523154
4  0.9233581   3.712419  0.2487214 0.80421371
6  5.8172905   3.115758  1.8670547 0.06647537
```

In addition to print and summary, a plot function for flimList is available in the package. Plotting a flimList object will display the standard lm plot diagnostics in a 2x2 grid for each time point. These plots enable us to detect any major misspecification in the linear models for the increments. Diagnostics for the increment model at time point 0 for the PANSS data are shown in Figure 3.

The top left plot is showing the residuals vs fitted values, and should ideally be close to zero and not show any trend. The plot in the top right is a Q-Q plot that can reveal deviation from the normal assumption in the linear regression. The bottom left plot is typically used in addition to the first plot to check for misspecification and to check for heterogeneity in variances – users should note however that it might not be a problem with heterogeneity as we do not have to assume homogeneity of variances (Diggle et al., 2007). The lower right plot that shows residuals vs leverage is used to detect if some observations are affecting the model fit more strongly than others, users should look for any observation with large leverage and a big standardised residual (negative or positive), as this may distort the model.

To view the plots for the subsequent time points, users are prompted to press <ENTER>:

```
> plot(flimList(panss.flim),"Y")
  Model: Y.inc ~ Y + factor(treat)
Hit <Return> to see next plot:
Time: 0
```

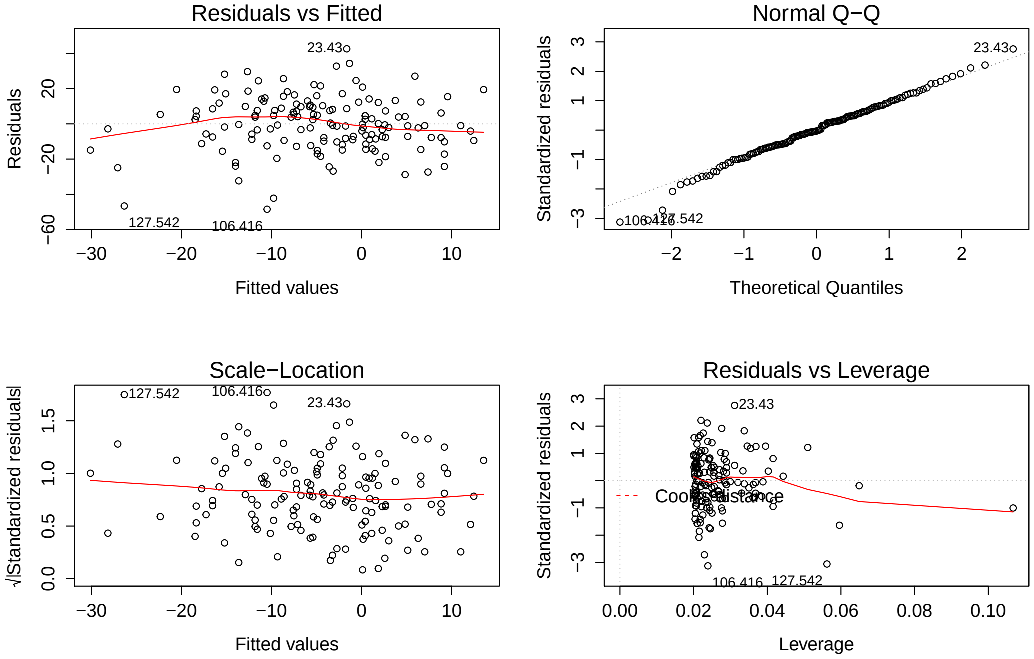

Figure 3: Plot diagnostics for linear fit at time point 0.

Bootstrapping

Bootstrapping is available once a "flim" object is fitted. Using the flimboot function, users have only to specify the number of bootstrap samples:

```
panss.boot <- flimboot(panss.flim, 100)
```

We can calculate the empirical standard errors for bootstrap estimates of the hypothetical mean response by:

```
> flimSD(panss.boot, "Y", grouping="treat")
```

To plot the hypothetical mean response from the original dataset together with point wise 95% bootstrap confidence bands, we can use the plot function for "flimboot" objects (Figure 4):

```
plot(panss.boot, "Y", "treat")
```

Artificial censoring and causal inference

As argued before, the linear increments model can be a potential method for doing causal inference. This is done by imputing counterfactual response values as if an intervention never took place; the imputed values can then be compared to the observed values to assess what effect the intervention had on the response.

We will demonstrate this feature with a simulated dataset that is made to mimic a study of patients with HIV. The patients in the study have steadily declining CD4 values, and as CD4 values continue to decrease, patients initiate treatment with increasing probability. Receiving treatment increases the patients' CD4 values. CD4 is also affecting the hazard of event as those with a low value are at greater risk of developing AIDS during the study. The event AIDS is simulated on each time point with an additive hazard model where CD4 status and treatment are affecting the hazard. If patients get AIDS, they are censored from the study.

Since CD4 is affecting both probability of starting treatment and the outcome, and treatment on the other hand is affecting CD4, it would be useful to know what the CD4 values would have been if patients counterfactually did not start treatment. Note that it is assumed that treatment is affecting the response value on the next observation after treatment starts, and then on consecutive observations. Once a patient starts treatment, he or she may not go back to being untreated.

It is simple to create counterfactual values as if treatment never started:

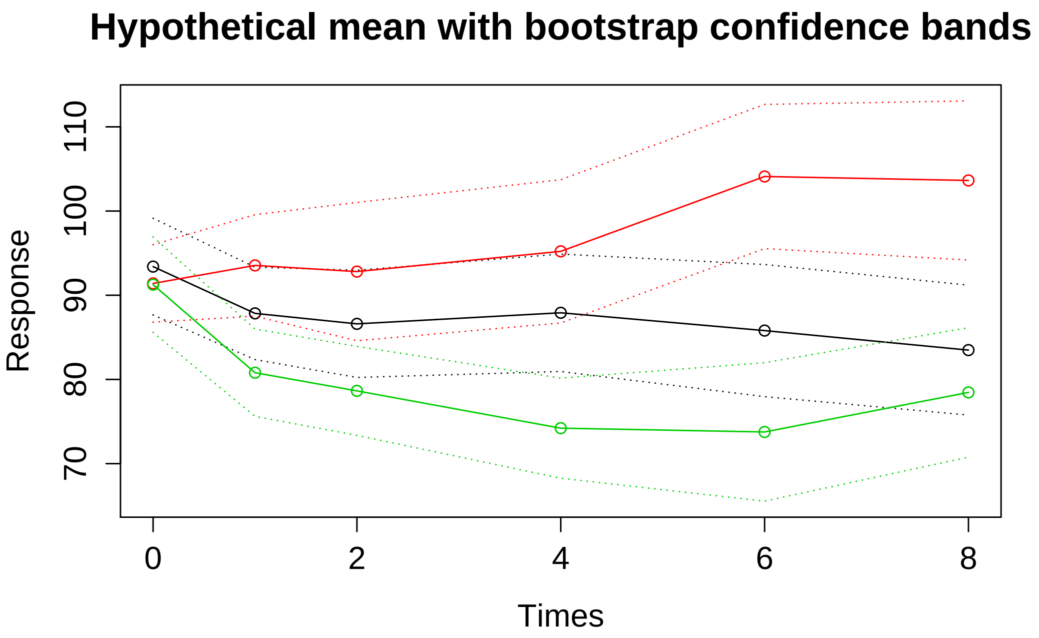

Figure 4: Hypothetical mean responses with bootstrap confidence bands.

```
data(CD4sim)
CD4.flim <- flim(cd4 ~ cd4, id="id", obstime="time", data=CD4sim,
             art.cens="treat")
```

The fitted object contains a dataset with the counterfactual CD4 values added in a separate column:

```
> head(CD4.flim$dataset)
  id time treat      cd4 AIDS   cd4.cf
1  1    0     0 7.071068    0 7.071068
2  1    1     0 6.084063    0 6.084063
3  1    2     0 5.471980    0 5.471980
4  1    3     0 4.644535    0 4.644535
5  1    4     0 3.900693    0 3.900693
6  1    5     0 3.262019    0 3.262019
```

This option is intended for advanced users and does not come with readily available tools for investigating results. In Figure 5 we have for illustration chosen to plot the mean response curves for observed and counterfactual CD4 values. Here we can see that the linear increments model suggests the CD4 values would have been lower if the treated patients were untreated.

Summary

The linear increments model is a simple approach for dealing with missing data in longitudinal studies, and FLIM provides a straightforward implementation with standard formula syntax. Using the package, we can recreate hypothetical response trajectories that could have been observed in the absence of missingness. The mean structure of the reconstructed data will be correct under certain assumptions (3); however, these imputed values will not exhibit the correct variability, and confidence bands for the mean response can be based on bootstrapping.

With a readily available tool for handling and imputing missing data, users should emphasise which assumptions need to be made in order for the hypothetical developments to be justifiable. Indeed, in some sense imputation is a manipulation of the data, and this is certainly true when one is considering averages such as the mean response. Uncritically applying the linear increments model to obtain a "complete" dataset is not advised.

Knowing when the required assumptions are met can be a challenge, and the issue is further complicated when missingness is nonmonotone. The basic message is that the method is applicable

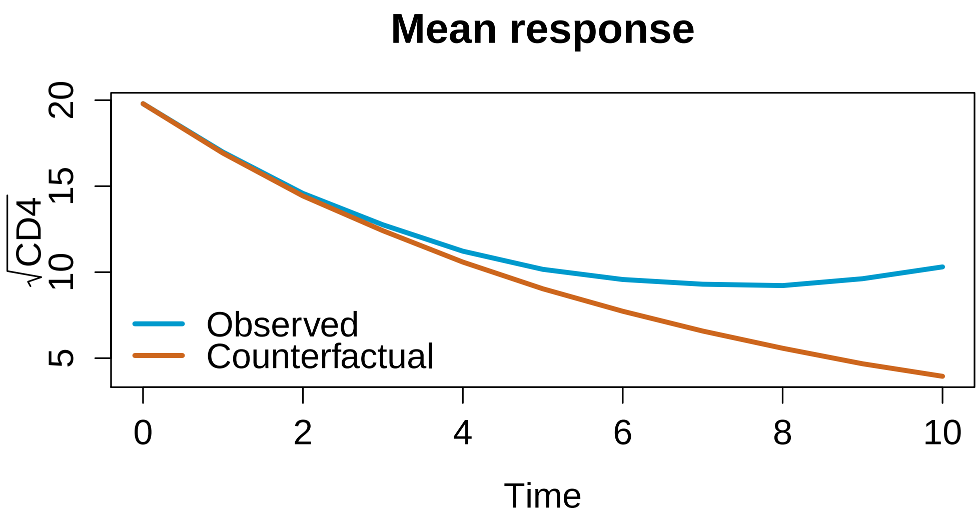

Figure 5: Mean response for observed and counterfactual CD4 values

when the observed increments are representative of those from the general population; this goes both for monotone and nonmonotone missing data, but may be more difficult to establish in the latter case.

Bibliography

O. O. Aalen and N. Gunnes. A dynamic approach for reconstructing missing longitudinal data using the linear increments model. *Biostatistics*, 11:453–472, 2010. [p137, 138, 139]

O. O. Aalen, K. Røysland, J. M. Gran, and B. Ledergerber. Causality, mediation and time: a dynamic viewpoint. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 175(4):831–861, 2012. [p141]

P. Diggle. Dealing with missing values in longitudinal studies. In *Recent Advances in the Statistical Analysis of Medical Data*, editor, B. S. Everitt and G. Dunn, pages 203–228, 1998. [p142, 143]

P. Diggle, D. M. Farewell, and R. Henderson. Analysis of longitudinal data with drop-out: objectives, assumptions and a proposal. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 56: 499–550, 2007. [p137, 138, 139, 146]

D. M. Farewell. *Linear models for censored data*. PhD thesis, Lancaster University, 2006. [p137]

N. Gunnes, D. M. Farewell, O. O. Aalen, and T. G. Seierstad. Analysis of censored discrete longitudinal data: estimation of mean response. *Statistics in Medicine*, 28:605–624, 2009. [p139]

M. A. Hernán, E. Lanoy, D. Costagliola, and J. M. Robins. Comparison of dynamic treatment regimes via inverse probability weighting. *Basic & Clinical Pharmacology & Toxicology*, 98:237–242 2006. [p140]

J. W. Hogan, J. Roy, and C. Korkontzelou. Handling drop-out in longitudinal studies. *Statistics in Medicine*, 23:1455–1497, 2004. [p137]

A. Zeileis and G. Grothendieck. zoo: S3 infrastructure for regular and irregular time series. *Journal of Statistical Software*, 14(6):1–27, 2005. URL http://www.jstatsoft.org/v14/i06/. [p142]

Rune Hoff
Oslo Centre for Biostatistics and Epidemiology
Department of Biostatistics
University of Oslo
Norway
rune.hoff@medisin.uio.no

Jon Michael Gran
Oslo Centre for Biostatistics and Epidemiology
Department of Biostatistics
University of Oslo
Norway
j.m.gran@medisin.uio.no

Daniel Farewell
Cochrane Institute of Primary Care & Public Health
School of Medicine
Cardiff University
Wales
FarewellD@cf.ac.uk