Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Broken Stick Model for Irregular Longitudinal Data

Stef van Buuren
Netherlands Organisation for Applied Scientific Research TNO & University of Utrecht

Abstract

Many longitudinal studies collect data that have irregular observation times, often requiring the application of linear mixed models with time-varying outcomes. This paper presents an alternative that splits the quantitative analysis into two steps. The first step converts irregularly observed data into a set of repeated measures through the broken stick model. The second step estimates the parameters of scientific interest from the repeated measurements at the subject level. The broken stick model approximates each subject's trajectory by a series of connected straight lines. The breakpoints, specified by the user, divide the time axis into consecutive intervals common to all subjects. We restrict the methodology to just three variables: time, measurement and subject. The model is a special case of the linear mixed model, with time as a linear B-spline and with subject as the grouping factor. The main assumptions are: Subjects are exchangeable, trajectories between two breakpoints are all straight, random effects follow a multivariate normal distribution, and unobserved data are missing at random (MAR). The **brokenstick** R package offers tools to calculate, predict, impute and visualise broken stick estimates. The package supports two optimisation methods, including options to constrain the variance-covariance matrix of the random effects. We demonstrate a few applications of the model: detection of critical periods, estimation of the time-to-time correlations, profile analysis, curve interpolation, multiple imputation and personalised prediction of future outcomes by curve matching.

Keywords: **brokenstick**, R, linear mixed model, repeated measures, linear B-spline, personalised estimation, growth curve analysis, critical periods, time-to-time correlation, profile analysis, curve interpolation, multiple imputation, curve matching, two-step method.

1. Introduction

Most longitudinal studies plan data collection to occur at a fixed set of time points. In practice, the realised times can differ - sometimes substantially - from the scheduled times. There may be many reasons for such differences. For example, we planned a visit in the

weekend or during a holiday, the subject didn't show up, the measurement device was out of order, or the investigator got ill. Varying observation times may also result from combining data from multiple studies, each collected according to its own design. Timing variation can be substantial in observational studies, especially if the survey lacks a pre-specified schedule. Longitudinal data with timing differences between subjects are said to be *irregular*.

Irregular observation times present significant challenges for quantitative analysis. For example, it isn't easy to calculate the time-to-time correlation matrix if the data spread thinly over time. It might also be complex to predict the future from past data if subject times differ. Observation times may also relate to the process of interest. For example, more severe patients get more frequent measurements; unmotivated cohort members respond more rarely, and so on. Conventional methods like MANOVA, regression or cluster analysis break down if observation times differ or if drop-out is selective.

While irregular observation times occur all over science, there is no universal or principled approach to resolve the problem. One straightforward fix is to take only those dates for which data are available (e.g., dates when stocks are traded), thus ignoring the times when markets are closed. One may also create bins of time intervals around the planned times, thereby ignoring within-period differences. Another ad-hoc method is to predict the value at the scheduled time from neighbouring data, e.g. by linear interpolation or smoothing, typically reducing the variability in the data. Some quick fixes create data sets where the timing problem seems to have "gone away", which may tempt the analyst to ignore the potential effects of data patch-up on the substantive conclusions. While convenient and straightforward, the thoughtless application of these fixes introduces significant spurious relations over time, especially if the spacing of observations is highly irregular.(Rehfeld, Marwan, Heitzig, and Kurths 2011) Binning can lead to "surprisingly large" biases.(Towers 2014) If timing variation is related to the outcome of interest, these methods may result in biased estimates and exaggerated claims.(Pullenayegum and Lim 2016)

The linear mixed model for longitudinal data (Laird and Ware 1982; Fitzmaurice, Laird, and Ware 2011) is the standard for the analysis of irregular data. The longitudinal mixed model represents each subject's observed curve by a parametric function of time. The parameter estimates of the function are specific to each subject and modelled as random effects. The linear mixed model is highly useful for irregular data since it borrows strength across different realisations of the same process, summarising each trajectory by a small number of parameters that vary over subjects. The analyst can break down the distribution of these random effects as a function of individual characteristics. The linear mixed model is attractive when the number of measurements differs between individuals, or when the measurements are taken at different times.

This paper explores the use of the *broken stick model* as a method to transform irregularly observed data into *repeated measures*. The broken stick model describes a curve by a series of connected straight lines. The model has a long history and is known under many other names, amongst others, *segmented straight lines* (Bellman and Roth 1969), *piecewise regression* (Toms and Lesperance 2003), *structural change models* (Bai and Perron 2003), *broken line smoothing* (Koutsoyiannis 2000) and *segmented regression* (Lerman 1980). The term *broken stick* goes back to at least MacArthur (1957), who used to it in an analogy to indicate the abundances of species. Most of the literature on the broken stick model concentrates on the problem of finding optimal times at which the lines should connect. Instead, the present paper will focus on the problem of summarising irregular individual trajectories by estimates made at

a *pre-specified time grid*. This time grid is identical for all individuals, but it needs not to be equidistant. Our model formulation is a special case of the linear mixed model, with time modelled as a set of random effects coded as a linear B-spline and with subjects as the grouping factor. The output of the transformation is a set of repeated measures, where every subject obtains a score on every time point.

Substantive researchers often favour repeated measures over the use of linear mixed models because of simplicity. For example, with repeated measures data, we can easily fit a subject-level model to predict future outcomes conditional on earlier data. While such simple regression models may be less efficient than modelling the full data (Diggle, Heagerty, Liang, and Zeger 2002, Sec. 6.1), increased insight may be more valuable than increased precision.

The broken stick model requires a specification of a sensible set of time points at which the measurements ideally should have been taken. For each subject, the model predicts or imputes hypothetical observations at those times. We apply the substantive analysis to the repeated measures instead of to the irregular data. This strategy is akin to Diggle's multi-stage approach model-fitting approach (Diggle 1988). The envisioned two-step analytic process aims to provide the best of both worlds.

Some applications of the broken stick model are:

- to approximate individual trajectories by a series of connected straight lines;
- to align irregularly observed curves to a common age grid;
- to impute realisations of individual trajectories;
- to estimate the time-to-time correlation matrix;
- to predict future observations.

My original motivation for developing the broken stick model was to facilitate the statistical analysis and testing of critical ages in the onset of childhood obesity (de Kroon, Renders, van Wouwe, van Buuren, and Hirasing 2010), with extensions to multiple imputation (van Buuren 2018). Anderson, Hafen, Sofrygin, Ryan, and HBGDki Community (2019) recommend the broken stick model because of good accuracy and ease of interpretation.

The present paper highlights various computational tools from the **brokenstick** package. The package contains tools to fit the broken stick model to data, to export the parameters of the fitted model for use outside the package, to create imputed values of the model, and to predict broken stick estimates for new data. Also, the text illustrates how the tool helps to solve various analytic problems.

2. Illustration of broken stick model

As a first step, let us study the variation in the age of measurement of 200 children from the SMOCC study (Herngreen, van Buuren, van Wieringen, Reerink, Verloove-Vanhorick, and Ruys 1994). Lokku, Lim, Birken, Pullenayegum, and TARGet Kids! Collaboration, (2020) suggest the *abacus plot* to visualise this variation.

The blue points in Figure 1 indicate the observation times. In general, the blue points are close to the scheduled ages (indicated by vertical lines), especially in the first half year. Observation times vary more for older children. Several children have one or more missing visits (e.g. 10002, 10008, 10024). Some children (10012, 10015) had fairly close visits. Child 10028 dropped out after month 9.

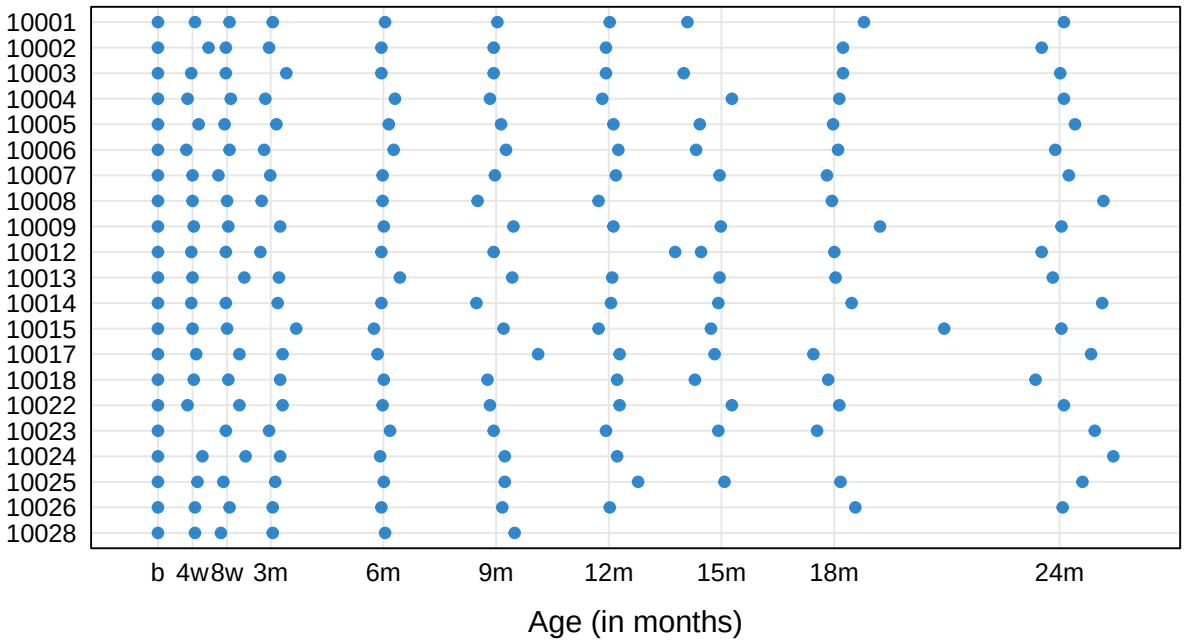

Figure 1: Abacus plot of observation times for the first 20 children of the SMOCC data.

Let us fit two models, with two and nine lines respectively, to the standard deviation score (SDS) of body length.

```
R> ids <- c(10001, 10005, 10022)
R> fit2 <- brokenstick(hgt.z ~ age | id, smocc_200, knots = 0:3)
R> knots <- c(0, 0.0833, 0.1667, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 2)
R> fit9 <- brokenstick(hgt.z ~ age | id, smocc_200,
+                      knots = knots, boundary = c(0, 3))
R> m2 <- plot(fit2, smocc_200, group = ids,
+            xlab = "Age (years)", ylab = "Length (SDS)")
R> m9 <- plot(fit9, smocc_200, group = ids,
+            xlab = "Age (years)", ylab = "Length (SDS)")
R> gridExtra::grid.arrange(m2, m9, nrow = 2)
```

Figure 2 shows the individual trajectories of three children. The blue points coincide with the observed data, whereas the red curves are calculated according to the broken stick model.

There are fitted two models. The simpler model (top) uses just two line segments. The first line starts at birth and ends at the age of exactly 1 years. The second line spans the period between 1 to 2 years. Note that the two lines connect at the breakpoint, the age of 1 year. The red curves for the two-line model are a crude approximation to the data.

We can create a better model by setting breakpoints equal to the scheduled ages. Since there are 10 scheduled ages, we construct nine straight lines. In contrast to the two-line model, the nine-line broken stick model is sensitive to small bumps in the observed trajectory and closely fits the empirical data. The residual variance of the nine-line model is low (0.059), and the proportion of explained variance in SDS is high, 0.98.

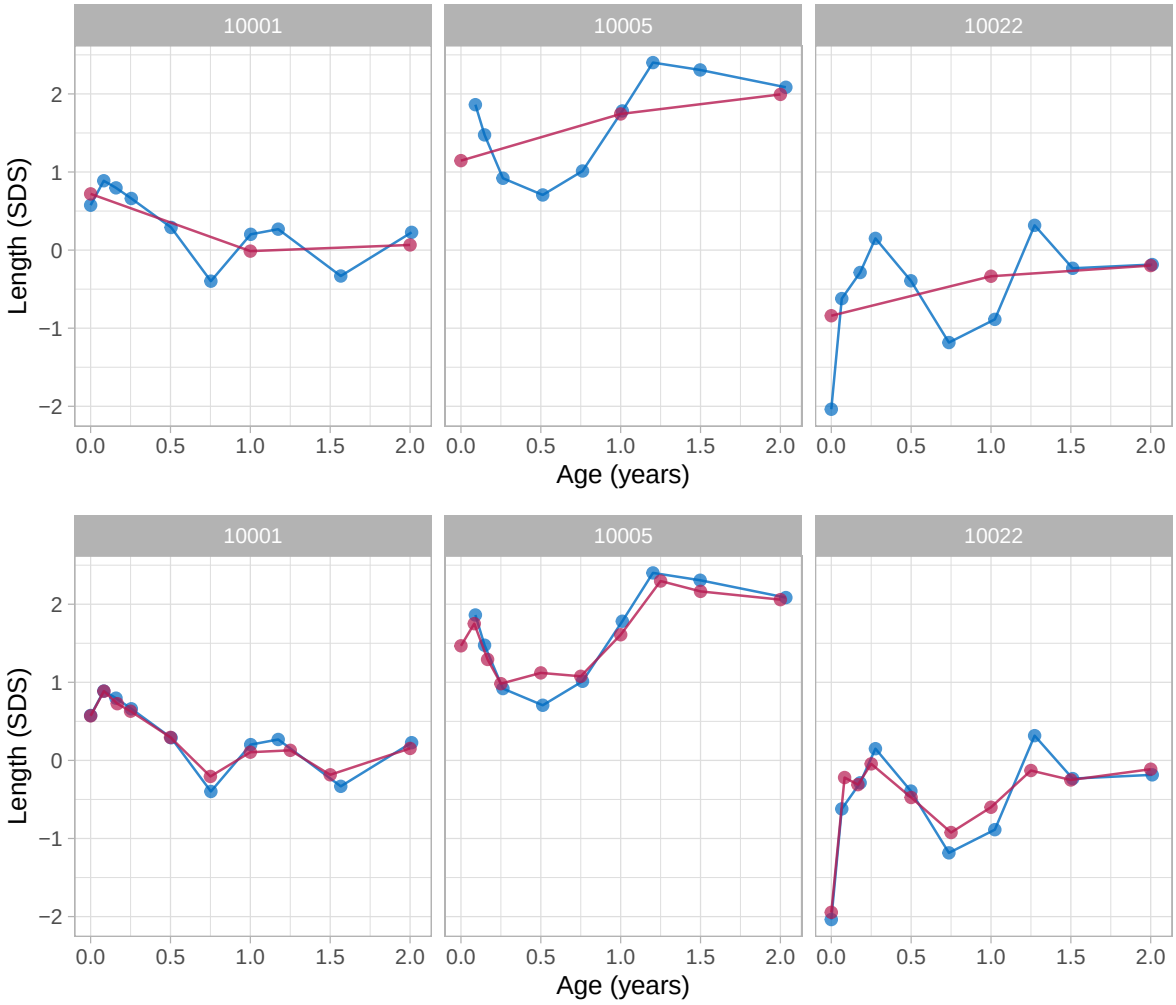

Figure 2: Broken stick model with two (top) and nine (bottom) line segments for three children. Blue = observed data, Red = Fitted broken stick curves.

While the observation times in the data differ between children, the broken stick curves use identical time points across subjects. The idea is now that we can add the broken stick estimates to the child-level data by a long-to-wide conversion, and analyse supplemented columns as repeated measures. A repeated measures analysis is usually simpler than the equivalent for the temporally misaligned data. For example, it is easy to calculate mean profiles for arbitrary groups, estimate the time-to-time covariance matrix or to build predictive models at the child level. See Hand and Taylor (1987) for a lucid overview of linear techniques for repeated measures.

3. Methodology

3.1. Notation

We adopt the notation of Fitzmaurice *et al.* (2011). Let Y_{ij} denote the response variable for the i^{th} subject on the j^{th} measurement occasion at time t_{ij}. Data are collected in a sample of N persons $i = 1, \ldots, N$. Let repeated measurements for the i^{th} subject be grouped as

$$Y_i = \begin{pmatrix} Y_{i1} \\ Y_{i2} \\ \vdots \\ Y_{in_i} \end{pmatrix}, \quad i = 1, \ldots, N.$$

If the measures have been observed at a common same set of occasions, then we could drop the index i in t_{ij} since $t_{ij} = t_j$ for all $i = 1, \ldots, N$. Here we will focus on the case that t_{ij} varies over i.

In addition, let use define the $n_i \times p$ matrices

$$X_i = \begin{pmatrix} X_{i11} & X_{i12} & \cdots & X_{i1p} \\ X_{i21} & X_{i22} & \cdots & X_{i2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{in_i1} & X_{in_i2} & \cdots & X_{in_ip} \end{pmatrix}, \quad i = 1, \ldots, N,$$

so that the rows of X_i contain p covariates associated with the responses at n_i measurement occasions. The columns may be time-varying covariates. If a certain covariate is fixed in time (e.g. sex, treatment, education), then all values within the corresponding column in X_i are identical.

3.2. Broken stick model

The broken stick model avoids modeling observation times t_{ij} directly by representing each t_{ij} as its relative position within a time interval. For example, suppose $t_{ij} = 0.6$ years and that the time interval is given by 0.5-1.0 years. The position relative to the left break age is $x_{\text{left}} = (1.0 - 0.6)/(1.0 - 0.5) = 0.8$, whereas relative to the right break age is $x_{\text{right}} = (0.6 - 0.5)/(1.0 - 0.5) = 0.2$. In order to fit the broken stick model, we need to replace time point $t_{ij} = 0.6$ by two values: 0.8 (for break age 0.5), and 0.2 (for break age 1.0). Note that both values add up to 1. Coding time in this way simplifies modeling continuous time by a set of discrete break ages.

More specifically, let t_{ij} be coded by a second-order (linear) B-spline using k internal knots κ placed at $k+1$ ordered ages

$$\kappa_0 = \kappa_1 < \cdots < \kappa_k < \kappa_{k+1}$$

The internal knots $\kappa_1, \ldots, \kappa_k$ correspond to the set of ages for which we obtain broken stick estimates, and it could be specified by the user. The left boundary knot $\kappa_0 = \kappa_1$ is left-anchored to the minimum time $\min(t_{ij})$ in the data. This point defines the starting event of the participant, such as birth or study enrolment. The right hand boundary knot is $\kappa_{k+1} \geq \max(t_{ij})$.

The second-order B-spline (de Boor 1978, pp. 32),

$$H_s(t) = \begin{cases} (t - \kappa_{s-1})/(\kappa_s - \kappa_{s-1}) & , \quad \kappa_{s-1} < t \leq \kappa_s, \\ (\kappa_{s+1} - t)/(\kappa_{s+1} - \kappa_s) & , \quad \kappa_s \leq t < \kappa_{s+1}, \\ 0 & , \quad \text{otherwise.} \end{cases}$$

is applied to t_{ij} to obtain $(k+1)$ transformed variables $x_{is} = t_{ij}$ with $s = 1, \ldots, k+1$. These variables can conveniently be grouped into the $n_i \times (k+1)$ matrix of covariates $X_i = (x_{i1}, \ldots, x_{ik}, x_{i(k+1)})$. Each row in X_i has only one or two non-zero elements, which sum to 1.

Using this X_i, the broken stick model is a special case (with $Z_i = X_i$) of the two-stage random-effects model (Laird and Ware 1982)

$$Y_i = X_i\beta + X_i b_i + \epsilon_i$$

where the $k+1$ column vector β contains $k+1$ fixed effect coefficients common to all persons, where the $k+1$ column vector b_i accomodates for $k+1$ subject-specific random parameters, and where the n_i column vector ϵ_i holds subject-specific residuals.

In order to complete the model specification, we assume that the residuals are identically and independently distributed as $\epsilon_i \sim N(0, \sigma^2 I(n_i))$, where σ^2 is a common variance parameter, and where $I(n_i)$ is the identity matrix of order n_i. Thus, the equation represents population parameters (fixed effects), individual effects (random effects), and an amount of within-person dispersion that is the same for all persons. The section on estimation also considers a heterogeneous model that allows σ_i^2 to vary over subjects.

In summary, given the knot specification and the choice of the response scale, the parameters of the broken stick model are:

- β, a vector of $k+1$ fixed parameters;
- Ω, a $(k+1) \times (k+1)$ covariance matrix of the random effects;
- σ^2, the within-person error variance.

The total number of parameters for a solution with k internal knots is thus equal to $(k^2+5k+6)/2$. For example, a model of $k = 3$ knots (i.e. with two connected lines) has 15 parameters, a model with $k = 4$ has 21 parameters, and a model with $k = 10$ break ages has 78 parameters.

3.3. Model assumptions

At the person level, we assume $b_i \sim N(0, \Omega)$, i.e., the random coefficients of the subjects have a multivariate normal distribution with zero mean and a $(k+1) \times (k+1)$ covariance

matrix Ω. The base model allows the elements of Ω to vary freely. For time-dependent data, constrained versions for Ω are also of interest.(Fitzmaurice *et al.* 2011, ch. 7). The estimation section highlights two such extensions. We also assume that the covariance between b_i and ϵ_i is zero. For simplicity, this paper is restricted to the case where X_i includes only time, and no other covariates. Also, we assume that X_i has no missing data.

The broken stick model builds upon three main modeling assumptions:

- The trajectory between break ages follows a straight line. This assumption may fail for processes that are convex or concave in time. For example, human height growth in centimeters growth is concave, so setting breakpoints far apart results introduces systematic model bias. Modeling height SDS instead of raw height will prevent this bias.
- The broken stick estimates follow a joint multivariate normal distribution. As this assumption may fail for skewed measurements, it could be beneficial to transform the outcomes so that their distribution will be closer to normal.
- The data are *Missing at Random* (MAR) given the outcomes from all subjects at all observation times. This assumption is restrictive in the sense that missingness may only depend on the observed outcomes, and not on covariates other than time. At the same time, the assumption is liberal in the sense that the missingness may depend on future outcomes. While this MAR-future assumption is unusual in the literature on drop-out and observation time models, it is a sensible strategy for creating imputations that preserve relations over time, especially for intermittent missing data. Of course, the subsequent substantive analysis on the imputed data needs to be aware of the causal direction of time.

3.4. Interpretation

Given the model estimates and the person data, we can calculate the random effect b_i. The *broken stick parameter* $\gamma_{is} = \beta_s + b_{is}$ is the subject-specific mean of Y_i at time κ_s, $s = 1, \ldots, k+1$. The set of γ_{is} parameters describes the mean response profile for subject i by k lines that connect at the $k+1$ coordinates (κ_s, γ_{is}).

The broken stick parameter is the most likely value of outcome Y_i for subject i at time κ_s. The parameter is the centre of the posterior predictive distribution for normal Y_i. The two-sided $100(1-\alpha)\%$ prediction interval for the true, though often unobserved, value Y_{i,κ_s} is equal to

$$[Y^{\text{lo}}_{i,\kappa_s}, Y^{\text{hi}}_{i,\kappa_s}] = \gamma_{is} \pm t_{(1-\alpha/2;N-1)}\sigma,$$

where $t_{(1-\alpha/2;N-1)}$ is the $100(1-\alpha/2)$ percentile of Student's t-distribution with $N-1$ degrees of freedom. For example, the 50% prediction interval $\gamma_{is} \pm 0.68\sigma$ will contain 50% of true values. For normal Y_i, the length of the 50% prediction interval is equivalent to the interquartile range (IQR). If the residual variation σ^2 is small (say $\sigma^2 < 0.1$), the IOR is about 0.22, so half of the true values will be within 0.22 SD of γ_{is}, a small difference. For large σ^2 (e.g. $\sigma^2 > 0.2$), the γ_i vector is a smoothed representation of Y_i. While smoothness amplifies low-frequency features of the trajectories, it could also introduce biases in the subsequent analysis by suppressing high-frequency variation. In that case, the analyst needs to check whether this reduction in variation does not affect the parameters of substantive interest. We

may restore high-frequency variation by adding random draws from the residual distribution $N(0, \sigma^2)$. From there, it is a small step to multiple imputation, a well-developed methodology for drawing valid inferences from incomplete data.(Rubin 1987; van Buuren 2018)

If $n_i >> k$ then the broken stick model provides a parsimonious representation of the measurements. Reversely, if $n_i << k$ then the model infers plausible values for subject i by building strength across persons. The broken stick model converts n_i irregularly observed measurements into a new set of k values γ_{is} at common ages $\kappa_1, ..., \kappa_k$, $s = 1, \ldots, k$.

Since each row in X_i sums to unity, the broken stick model does not have a global intercept. The linear B-spline coding effectively replaces the global random intercept term by $k+1$ *local intercepts*, one at each break age. The local intercept summarizes the information available in the adjacent left and right age intervals and ignores any information beyond the two adjacent knots. The broken stick estimates are thus primarily local. Outcome data observed outside the two adjacent age intervals influence the broken stick estimates only through the subject-level part of the model, in particular through Ω.

3.5. Estimation

Parameter estimation, method lmer

Estimation of the broken stick model relies on two well-developed R functions: `splines::bs()` (R Core Team 2020) and `lme4::lmer()`.(Bates, Mächler, Bolker, and Walker 2015) The following snippet illustrates how the `brokenstick::make.basis()` function calculates the matrix of B-splines for the time variable `age`:

```
R> library(splines)
R> data <- brokenstick::smocc_200
R> brk <- c(0, 0.5, 1, 2)
R> X <- bs(data$age, knots = brk, Boundary.knots = c(0, 3), degree = 1)
R> colnames(X) <- paste("age", c(brk, 3), sep = "_")
R> data <- cbind(data[, c("id", "age", "hgt.z")], X)
R> head(data)
```

```
     id   age hgt.z age_0 age_0.5  age_1 age_2 age_3
1 10001 0.000  0.57  1.00    0.00 0.0000     0     0
2 10001 0.082  0.89  0.84    0.16 0.0000     0     0
3 10001 0.159  0.80  0.68    0.32 0.0000     0     0
4 10001 0.255  0.66  0.49    0.51 0.0000     0     0
5 10001 0.504  0.29  0.00    0.99 0.0076     0     0
6 10001 0.753 -0.40  0.00    0.49 0.5058     0     0
```

The numerical example shows that the `bs()` function transforms the `age` variable into five columns, the B-spline basis, with names like `age_0` and `age_0.5`. If `age` coincides with one of these (e.g., as in the top row), then the corresponding column receives a `1`. In all other cases, age distributes over two adjacent columns. To make things fit, we need an additional column (here `age_3`) at the last position, the right boundary knot. There is also a left boundary knot, and I have conveniently set that equal to the first breakpoint, marking the start of

time. Setting `degree = 1` specifies a B-spline gives the broken stick model its name and its characteristic shape.

To illustrate the second step of the calculations, we call `lme4::lmer()` as follows:

```
R> library(lme4)
R> ctl <- .makeCC("warning", tol = 4e-3)
R> f <- hgt.z ~ 0 + age_0 + age_0.5 + age_1 + age_2 + age_3 +
+   (0 + age_0 + age_0.5 + age_1 + age_2 + age_3 | id)
R> fit <- lmer(f, data,
+              control = lmerControl(check.conv.grad = ctl))
R>
R> ### fitted trajectories for all persons
R> bse <- t(t(ranef(fit)$id) + fixef(fit))
R> head(round(bse, 3), 3)
```

```
      age_0 age_0.5 age_1  age_2 age_3
10001  0.78    0.27 -0.01  0.076  0.11
10002 -0.28   -0.21 -0.46 -0.478 -0.10
10003  1.68    1.97  1.28  1.115 -0.89
```

The broken stick estimates are the sum of the fixed and random effects. We need to remove the intercept as predictor rows all sum to 1. Warnings often occur when fitting broken stick models with `lme4::lmer()`. Here we have surpressed the warning `## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model failed to converge with max|grad| = 0.0031397 (tol = 0.002, component 1)` by setting `tol` to a more relaxed value. I have found that the broken estimates still look sound and reasonable. Most of my experience derives for child growth data, so there is no guarantee that this apparent robustness will hold for other types of data. Warnings become less frequent for a lower number of breakpoints and larger samples sizes.

The **brokenstick** package can fit the same model in an easier way:

```
R> ctl <- control_brokenstick(
+   lmer = lmerControl(check.conv.grad =
+                        .makeCC("warning", tol = 4e-3)))
R> mod <- brokenstick(hgt.z ~ age | id, data = smocc_200,
+                     knots = c(0, 0.5, 1, 2), boundary = c(0, 3),
+                     control = ctl)
R> head(predict(mod, smocc_200, x = "knots", shape = "wide"), 3)
```

```
# A tibble: 3 x 6
     id    `0`  `0.5`      `1`     `2`    `3`
  <dbl>  <dbl>  <dbl>    <dbl>   <dbl>  <dbl>
1 10001  0.778  0.272 -0.00986  0.0760  0.107
2 10002 -0.278 -0.206 -0.460   -0.478  -0.101
3 10003  1.68   1.97   1.28     1.11   -0.892
```

Parameter estimation, method kr

The calculation time of `lme4::lmer()` rapidly increases with the number of random effects. More than ten random effects (knots) takes significant time, and beyond 15 knots is generally impossible to fit. The **brokenstick** package provides another alternative, the *Kasim-Raudenbush (KR) sampler* (Kasim and Raudenbush 1998), which simulates draws from the posterior distributions of parameters from a two-level normal model with heterogeneous within-subject variances. The speed of the KR-Raudenbush sampler is almost insensitive to the number of random effects and depends primarily on the *total number of iterations*. The `brokenstick::kr()` function provides some reasonable defaults. The behaviour of the method has not been studied as well as `lmer()` and should still be considered experimental.

Apart from being faster, the KR-sampler opens up interesting analytic options:

1. It is relatively easy to constrain the fitted covariance of random effects, Ω, to a matrix of simple structure. Informing the sampler of the time-dependent structure of the random effect leads to stabler estimates of Ω. The package currently implements two *correlations models*. These models express the correlation $\rho(t_1, t_2)$ between two Z -scores Z_1 and Z_2 at successive ages t_1 and t_2 as a function of those ages. The *Argyle model* (Argyle, Seheult, and Wooff 2008) is $\rho(t_1, t_2) = \exp(-\lambda|T_1 - T_2|)$, where $T_i = \log(\tau + t_i)$ is a logarithmic rescaling of the time axis and $\rho = exp(-\lambda)$. The Cole correlation model (Cole 1995) describes the Fisher-transformed correlation as a function of the average $(t_1 + t_2)/2$ and the difference $(t_2 - t_1)$, including two multiplicative terms. Note that both models were proposed in the context of child growth, and have not been tested for other types of time-dependent data.

2. The KR-sampler fits the slightly more general linear-mixed model with heterogeneous within-subject variances, i.e. with a residual variance σ_i^2 per subject i instead of the global residual σ^2. This makes it easier to identify, study and weight subjects based on how well they fit the model.

3. A third option is to simulate imputations as an extra step to the KR-sampler. For subject with large σ_i^2, the random effect estimates are a too smooth representation of the data, leading to inappropriate variance estimates when those estimates are analysed as "just data". Section 11.3 of van Buuren (2018) pioneered a solution that constructs multiple trajectories by adding a proper amount of residual noise to random effect estimates. The variance estimation then proceeds according to the principles of multiple imputation.(Rubin 1987)

Random effects estimation

Apart from parameter estimates of the broken stick model, we also need a way to estimate random effects for a given set of model estimates and (new) user data. There are several of such methods. The `brokenstick::EB()` function implements the empirical Bayes (EB) estimate, also known as BLUP (Skrondal and Rabe-Hesketh (2009), p. 683). The procedure can provide the broken stick estimates for new persons. It the workhorse of the more user-

friendly `predict()` method that takes fitted models objects.

4. Functionality

4.1. Overview of brokenstick package

The **brokenstick** package contains functions to fit, predict and plot data. The main functions in the `brokenstick` package are:

Function name	Description
`brokenstick()`	Fit a broken stick model to irregular data
`predict()`	Predict broken stick estimates for new data
`plot()`	Plot individual trajectories

The following functions are user-oriented helpers:

Function name	Description
`fitted()`	Calculate fitted values
`get_knots()`	Obtain the knots used by model
`get_r2()`	Obtain proportion of explained variance
`residuals()`	Extract residuals from model

The following functions are responsible for calculations:

Function name	Description
`control_brokenstick()`	Set controls to steer calculations
`EB()`	Empirical Bayes predictor for random effects
`kr()`	Kasim-Raudenbush sampler for two-level model
`make_basis()`	Create linear splines basis

The package follows the **tidymodels** conventions. For example, the modelling object does not store the training data, whereas the convention dictates the variable names. The package architecture borrows important ideas from the **hardhat** package.(Vaughan and Kuhn 2020)

4.2. Data preparation

Before we can fit the model, the data need to be in shape. Data preparation is often the most time-consuming part of the analysis. The `brokenstick()` function takes tidy data in the long-form, with every observed subject-time combination in a row. This section uses the built-in `smocc_200` data, containing the heights of 200 children measured at ten visits up to two years.(Herngreen *et al.* 1994)

```
R> library(brokenstick)
```

```
R> head(smocc_200, 3)

# A tibble: 3 x 7
     id    age sex       ga    bw   hgt hgt.z
  <dbl>  <dbl> <chr>  <dbl> <dbl> <dbl> <dbl>
1 10001 0      female    40  3960  52   0.575
2 10001 0.0821 female    40  3960  55.6 0.888
3 10001 0.159  female    40  3960  58.2 0.797
```

4.3. Calculate Z-scores

The broken stick model can fit observations in either the raw scale (cm, kg, and so on) or as a standard deviation score (SDS), or Z-score. The results from the analysis of the Z-score is preferable for several reasons:

1. for growth curves, a straight line assumption is more plausible in the Z-score scale;
2. observations in the Z-score scale are closer to multivariate normality;
3. analysis of Z-scores highlights the interesting variation within and between children;
4. fitting Z-score data leads to fewer convergence issues.

It is easy to convert the measurements into the Z-score scale, fit the model, and convert back to the raw scale afterwards, if desired. There are several R packages that assist in the calculations: **AGD**, **anthro**, **childsds**, **growthstandards** and **zscorer**.

The `smocc_200` data contains the height measurement both in the original scale in cm (`hgt`) and the Z-score scale (`hgt.z`) relative to the height references from the Fourth Dutch Growth study (Fredriks, van Buuren, Burgmeijer, Meulmeester, Beuker, Brugman, Roede, Verloove-Vanhorick, and Wit 2000). Let us recalculate height SDS using the **AGD** package. Fortunately, we find that the same values.

```
R> library(AGD)
R> z <- with(smocc_200, y2z(y = hgt,
+                           x = age,
+                           sex = ifelse(sex == "male", "M", "F"),
+                           ref = nl4.hgt))
R> identical(z, smocc_200$hgt.z)

[1] TRUE
```

Figure 3 shows that, as expected, the empirical Z-score distribution is close to the standard normal. The few very extremely low heights correspond to pre-term born infants. The next section concentrate on modelling `hgt.z`.

Function `z2y()` applies the inverse transformation of Z-scores to the original scale. The following snippet converts `hgt.z` into the cm scale.

```
R> y <- with(smocc_200, z2y(z = hgt.z,
+                           x = age,
```

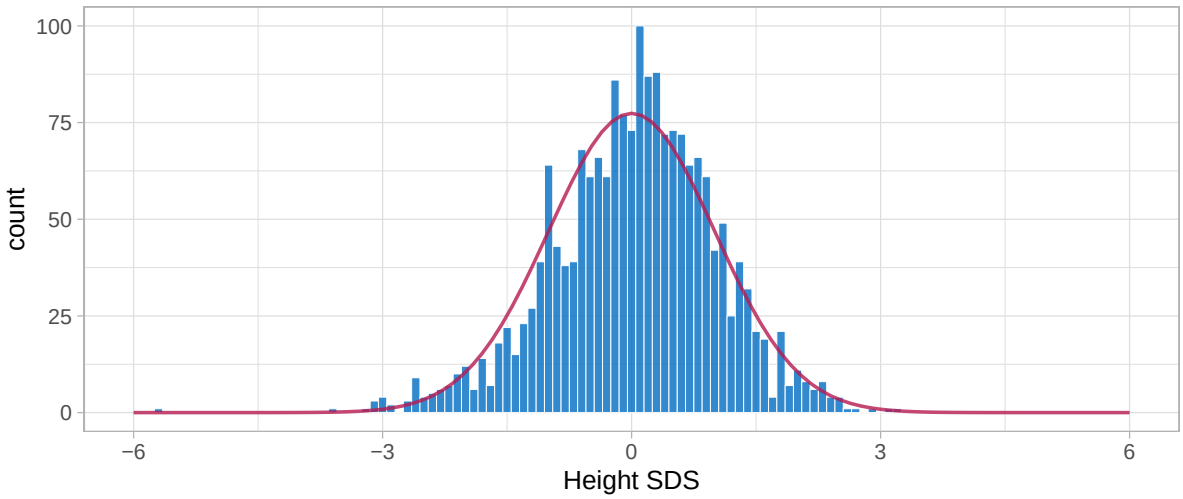

Figure 3: Distribution of height SDS for 200 Dutch children.

```
+                          sex = ifelse(sex == "male", "M", "F"),
+                          ref = nl4.hgt))
R> all.equal(y, smocc_200$hgt, tol = 0.0001)

[1] TRUE
```

We have used the Dutch 1997 height references here, but there are more choices. The **AGD** package also supports the WHO Child Growth Standards.

In practice we found that the model fit is often better when applied to Z-scores. Age-conditional references are common in child growth exists, but could be rare in other fields. An alternative is to apply the broken stick model to the standardized residuals of a preliminary non-linear regression of the outcome on time.

4.4. Model fitting

Figure 4 displays the growth curves of a subset of 52 children. The Z-score transformation takes away the major time trend, so all trajectories are more or less flat. This display allows us to see an extremely detailed assessment of individual growth. Note how the measurements cluster around ten ages: birth, 1, 2, 3, 6, 9, 12, 15, 18 and 24 months. While the data collectors rigorously followed the study design, variation in timing is inevitable because of weekends, holidays, sickness, and other events.

Fit one line

As a start, let us fit a simple model with just one line anchored at the minimum and maximum age.

```
R> fit <- brokenstick(hgt.z ~ age | id, smocc_200)
R> ids <- c(10001, 10005, 10022)
R> plot(fit, new_data = data, group = ids, what = "all",
+      xlab = "Age (years)", ylab = "Length (cm)")
```

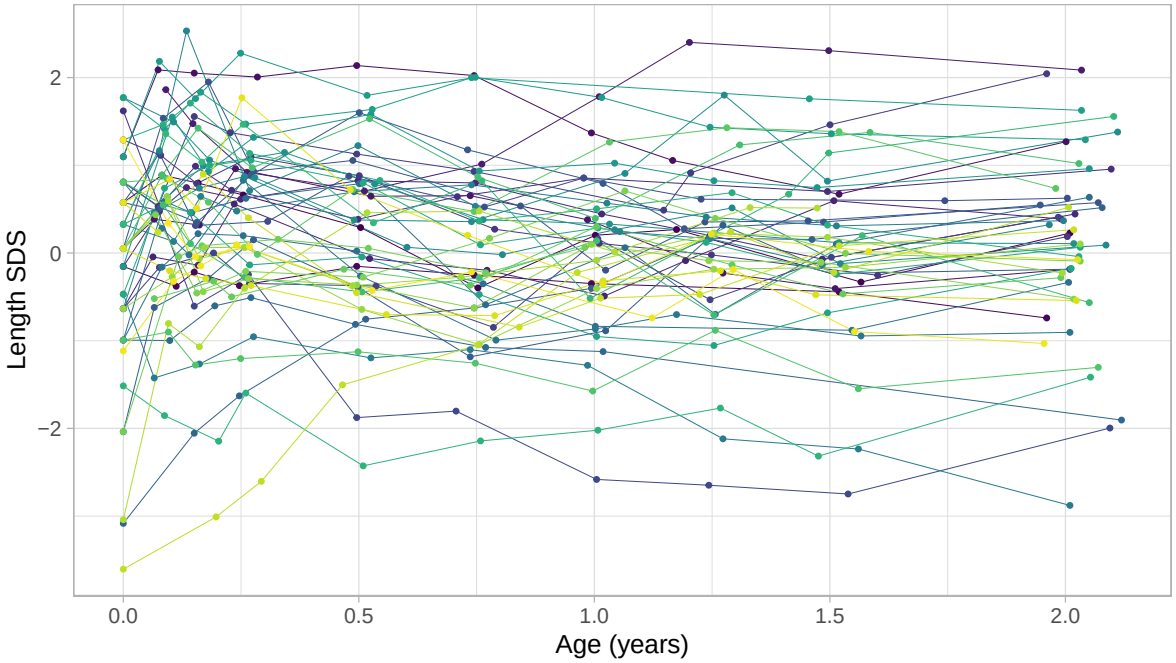

Figure 4: Length growth of 52 infants expressed in the Z-score scale.

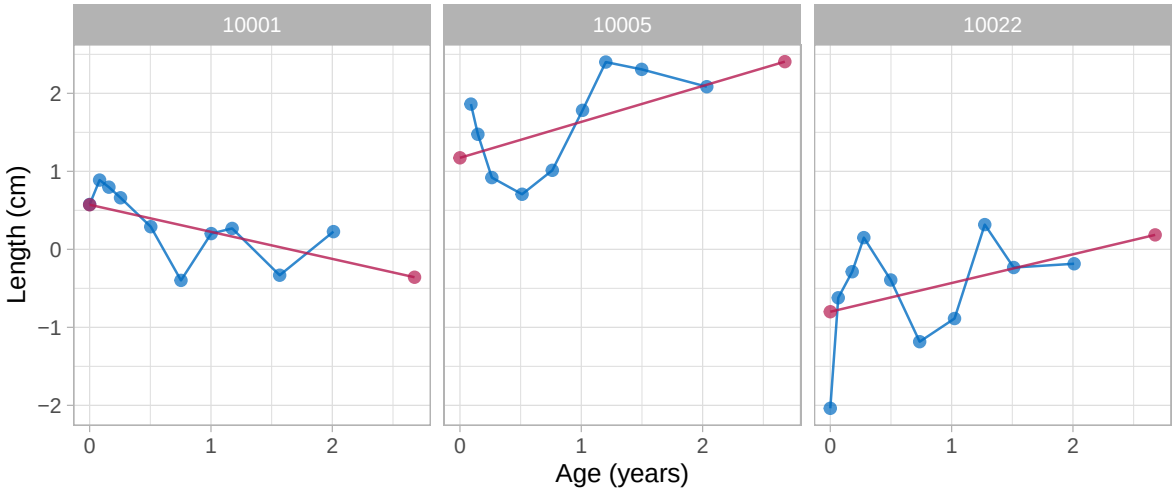

Figure 5: Simple linear model with one line anchored at the extremes.

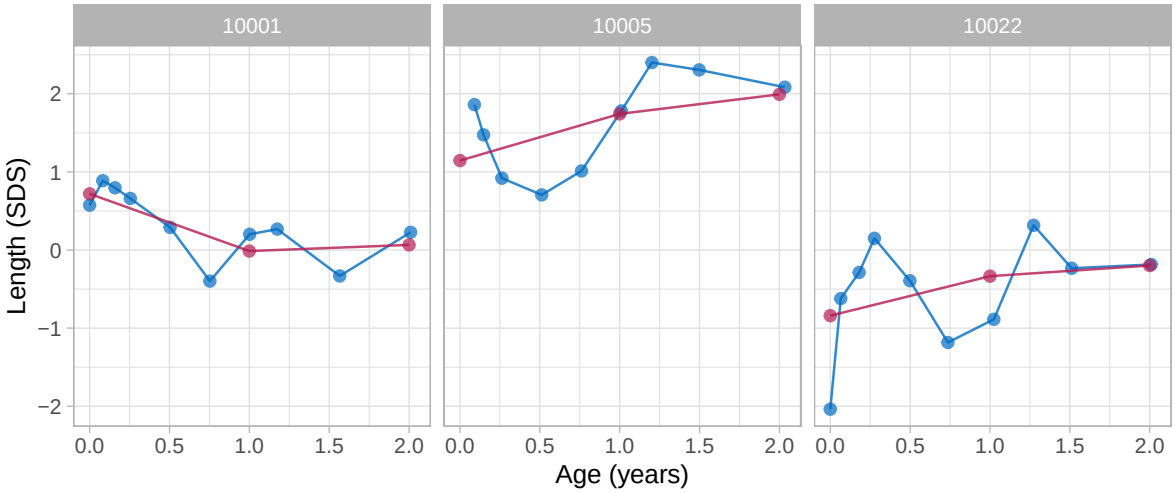

Figure 6: Broken stick model with two lines.

Figure 5 shows the observed (blue) and fitted (red) trajectories of three selected children. Note that this model can only capture the overall age trend. As a result, the approximation to the data is quite bad.

Fit two lines

We now extend to two connected lines. The first line should start at birth and end at the age of one year. The second line spans the period between one to two years. The lines must connect at the age of one year. We estimate and plot the model as follows:

```
R> fit2 <- brokenstick(hgt.z ~ age | id, smocc_200, knots = 0:2)
R> plot(fit2, data, group = ids,
+      xlab = "Age (years)", ylab = "Length (SDS)")
```

The `fit2` object holds the parameter estimates of the model:

```
R> fit2
```

```
Class: brokenstick (NULL)
Knots: 0 1 2 2.7
Means: -0.05 0.03 0.07 0.15
Variance-covariance matrix:
           age_0 age_1 age_2 age_2.6776
age_0       1.19
age_1       0.46   0.8
age_2       0.47  0.76  0.83
age_2.6776 -0.13  0.12  0.27       0.33
Residual variance:  0.18
```

The printed output lists the knots of the model at 0, 1, 2 and 2.6776 years. The left and right boundaries are located at 0 and 2.6776, respectively. The `means` entry lists the fixed

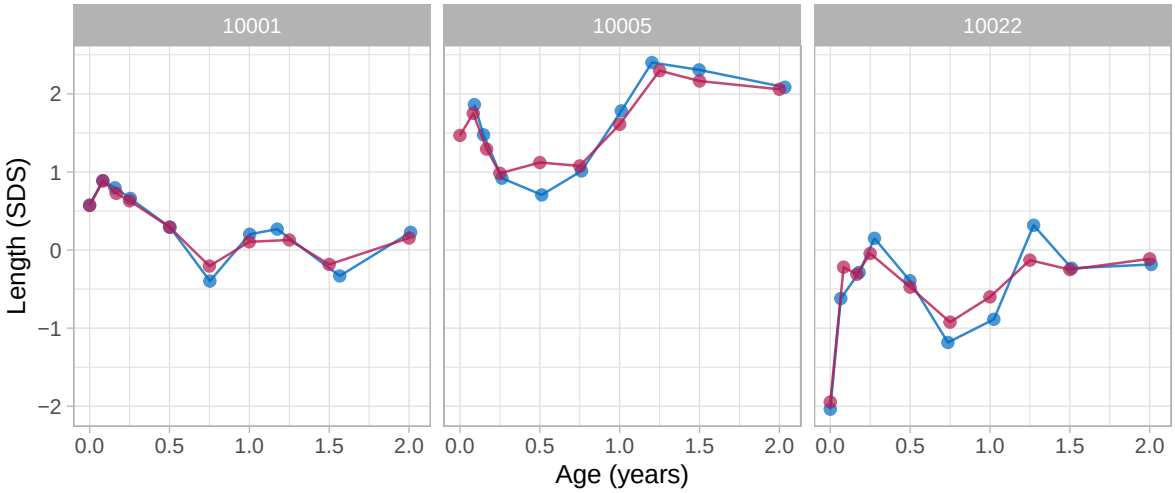

Figure 7: Broken stick model with nine lines.

effect estimates, which we interpret as the average SDS per time point. The time-to-time variance-covariance matrix cover four random effects (3 visits + 1 end knot). The residual variance measures the variability of the discrepancies between the model and the observed data. These three parameters (fixed, random, residual variance) are well interpretable and fully record the fitted broken stick model.

Fit nine lines

The two-line model does not fit well. We substantially refine the model by adding a knot for each scheduled visit. To make model specification independent of the data, we specify the right boundary as a constant of three years. We code and run the model as

```
R> knots <- round(c(0, 1, 2, 3, 6, 9, 12, 15, 18, 24)/12, 4)
R> fit9 <- brokenstick(hgt.z ~ age | id, data = smocc_200,
+                      knots = knots, boundary = c(0, 3))
```

This optimization problem is more complicated and time-consuming. As noted before, it is common for the optimization software to issue warnings, often related to the number of random effects relative to the number of observations. While these may be a little discomforting, we have found that the warnings are generally at the conservative side, and that the parameter estimates seem OK. With a small residual variance of 0.059, the nine-line broken stick model fits the observed data very well.

The training set includes all subjects. Depending on the study goals, we may wish to further improve the model fit by removing children from the data. For example, there might be children for which few observations are available, children with diseases, or children with trajectories that are very unusual or faulty. Be aware that such removals preserve the external generalisability of the training sample.

4.5. Prediction

Once we have a fitted model, we may obtain predictions. The subject(s) could be part of the training sample, but could also consist of new children.

All subjects

The `predict()` function obtains predictions from the broken stick model. The function is flexible, and allows for prediction of new subjects at arbitrary ages in a variety of output formats. The simplest call

```
R> p1 <- predict(fit9, smocc_200)
R> head(p1, 3)
```

```
  .pred
1  0.57
2  0.88
3  0.74
```

produces the predicted value (in `.pred`) for each row in `data`.

The predicted values represent a compromise between the person's data values and the global mean. In general, the fewer and less extreme data points of a person are, the closer the compromise will be toward the global mean. The compromise is called the *conditional mean* of the posterior distribution, the sum of the fixed and random effects.

We can obtain the locations at which the lines connect by specifying the `x = "knots"` argument, e.g.

```
R> p2 <- predict(fit9, smocc_200, x = "knots")
R> head(p2, 3)
```

```
# A tibble: 3 x 9
  .source    id    age sex      ga    bw   hgt hgt.z .pred
  <chr>   <dbl>  <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
1 added   10001 0      <NA>     NA    NA    NA    NA 0.571
2 added   10001 0.0833 <NA>     NA    NA    NA    NA 0.886
3 added   10001 0.167  <NA>     NA    NA    NA    NA 0.726
```

This is case 1 in the help of `predict.brokenstick()`. The result `p2` is a table with 2200 rows (= 11 knots × 200 subjects). The rows include additional identifying information. Adding the `shape = "wide"` argument transforms the information into *repeated measures*, with 200 rows and $12 = 1 + 11$ columns, that form supplemental variables for further analyses at the subject level.

We may also obtain both the conditional means as well as predictions at the observation ages for all children by

```
R> p3 <- predict(fit9, smocc_200, x = "knots", strip_data = FALSE)
R> head(p3, 3)
```

```
# A tibble: 3 x 9
  .source    id    age sex       ga    bw   hgt hgt.z .pred
  <chr>   <dbl>  <dbl> <chr>  <dbl> <dbl> <dbl> <dbl> <dbl>
1 data    10001 0      female    40  3960  52   0.575 0.571
2 data    10001 0.0821 female    40  3960  55.6 0.888 0.882
3 data    10001 0.159  female    40  3960  58.2 0.797 0.742
```

which contains 4140 rows (= 1940 data points + 2200 added points).

Now suppose that we desire to predict height SDS at other ages, e.g. at 0.42, 1.33 and 4 years. We can do so by (case 4, all groups)

```
R> head(predict(fit9, smocc_200, x = c(0.42, 1.33, 4), shape = "wide"), 3)

# A tibble: 3 x 4
     id `0.42`  `1.33`   `4`
  <dbl>  <dbl>   <dbl> <dbl>
1 10001  0.402  0.0287    NA
2 10002 -0.265 -0.488     NA
3 10003  2.06   1.04      NA
```

Thus, we have some flexibility to work with times that are not breakpoints. Remember though that the underlying model did not change. For example, we cannot magically predict outside the model at age 4.

Single subject

Obtaining predicted values per subject requires the `group` argument (case 3). For example

```
R> predict(fit9, smocc_200, group = 10001, shape = "vector")

 [1]  0.57  0.88  0.74  0.62  0.29 -0.20  0.11  0.12 -0.14  0.16
```

returns the vector of predictions for child 10001. Remove the `shape` argument to append the child's data. Also, here we can predict at other times using the `x` argument (case 4).

Now suppose that for subject 10001 we have additional height data at ages 0.42 and 1.33 years. Can we predict the child's trajectory with these new points included? The answer is yes. The command (case 5)

```
R> tail(predict(fit9, smocc_200, x = c(0.42, 1.33), y = c(-0.5, -1),
+               group = c(10001, 10001), strip_data = FALSE), 3)

# A tibble: 3 x 9
  .source    id   age sex       ga    bw   hgt  hgt.z   .pred
  <chr>   <dbl> <dbl> <chr>  <dbl> <dbl> <dbl>  <dbl>   <dbl>
1 data    10001  2.01 female    40  3960  88.3  0.227  0.0687
2 added   10001  0.42 <NA>      NA    NA  NA   -0.5    0.217
3 added   10001  1.33 <NA>      NA    NA  NA   -1     -0.254
```

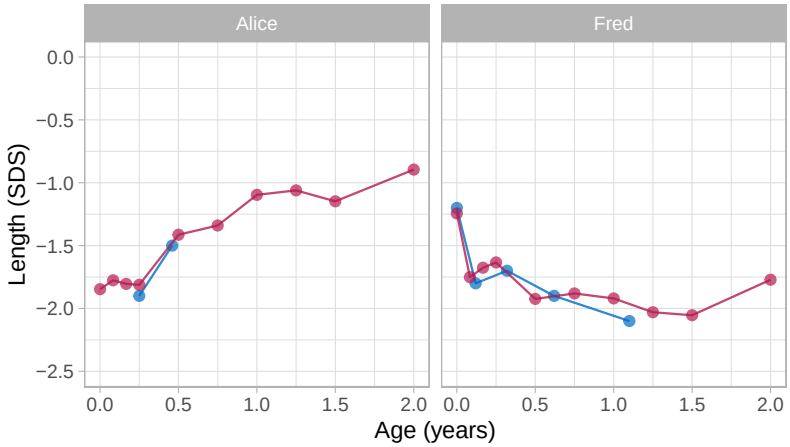

Figure 8: Alice and Fred - observed (blue) and fitted (red) trajectory.

appends two new records to the data of child 10001, and recalculates the trajectory using all data.

New subject

Suppose we have measured two children, Fred and Alice. We wish to obtain predictions for both using the model `fit9`. The following snippet calculates predictions at both the observed ages and at the knot locations:

```
R> data <- data.frame(
+   age = c(0, 0.12, 0.32, 0.62, 1.1, 0.25, 0.46),
+   hgt.z = c(-1.2, -1.8, -1.7, -1.9, -2.1, -1.9, -1.5),
+   id = c(rep("Fred", 5), rep("Alice", 2)))
R> p <- predict(fit9, data, x = "knots", strip_data = FALSE)
```

We can plot the trajectories data by

```
R> plot(fit9, data, ylim = c(-2.5, 0), xlab = "Age (years)", ylab = "Length (SDS)")
```

Alice contributes only two data points in the first half-year. The model expects that her height SDS will be around -1 SD at the age of two years. Using the data up to 1.1 years, the model predicts that Fred's growth curve remains around -2.0 SD until Fred is 1.5 years, and then increases to around -1.8 SD. While both predicted trajectories are extreme extrapolations, the example illustrates that it is possible to make informed predictions using just a handful of data points.

The `predict()` function does not care about whether `new_data` is the training data or not. All options are supported in both training and test data, thus providing ample flexibility to suit many use cases.

4.6. Quality of prediction

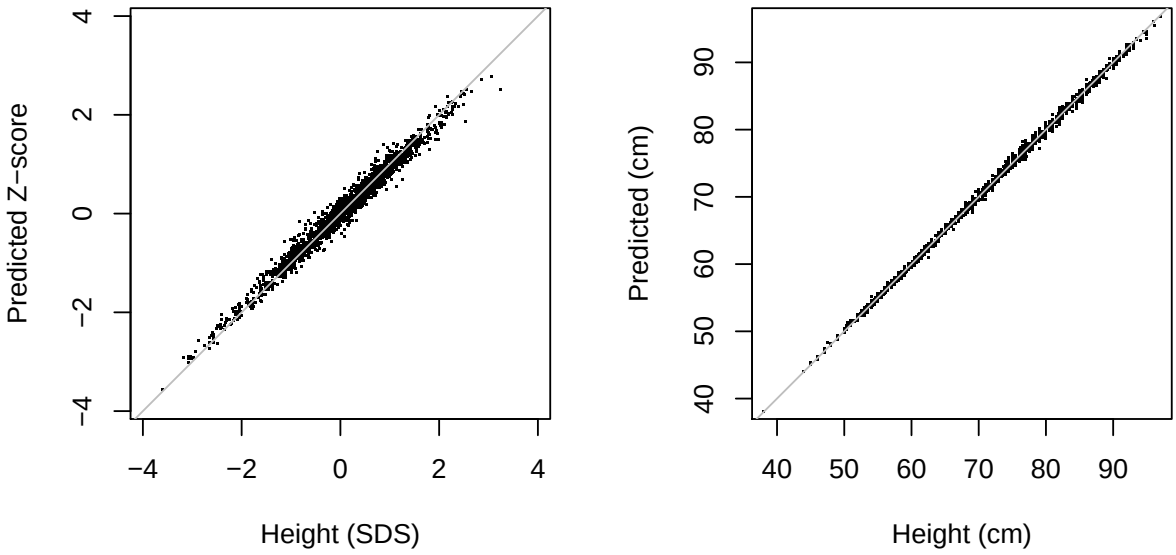

Figure 9: Predicted versus observed values.

Figure 9 is the scatterplot of the observed versus predicted values provides a visual representation of the accuracy of the prediction of the model in height SDS and cm scales. Both plots suggest an excellent fit between the observed and fitted data. The percentage of explained variance for the height SDS is high: 97.8%. The standard deviation of the residuals is equal to 0.152 SD, a small value in the Z-scale. When back-converted to centimetres, the scatterplot of the observed versus predicted values is even a little tighter. The proportion of explained variance is close to perfection: 99.9%. The standard deviation of the residuals is 4 mm, about the size of the technical error of measurement (TEM) for duplicate measurements in infants.(Ismail, Puglia, Ohuma, Ash, Bishop, Carew, Al Dhaheri, and Chumlea 2016, Table 2)

The model is as good as it can get. The uncertainties associated with the transformation from varying observation times to repeated measures will be small. For all practical purposes, the results from a linear mixed or multilevel model and a repeated measures model are likely to be same.

4.7. Knot placement strategies

Fitting the broken stick model requires a specification of the knots. The choice of the knots influences the quality and usefulness of the solution, so exercise some care in setting appropriate knot locations.

The `brokenstick()` function uses the same set of knots for all subjects. By default, the procedure places the boundary knots at the range of the data and no inner knots, resulting in a model that is linear in time without breakpoints. The `k` argument is a quick way to add `k` internal knots at equidense quantiles of the time variable. For example, specifying `k = 1` puts a knot at the 50th quantile (median), setting `k = 3` puts knots at the 25th, 50th and 75th quantiles, and so on. While convenient and quick, this option can result in suboptimal knot placement that is not adequate for the problem at hand. In general, it is best to specify explicit values for the `knots` and `boundary` arguments.

Here are some suggestions for knot placement:

1. If you want to predict at specific ages, then specify knots at those ages. For example, if the scientific interest includes prediction at the age of 1 and 2 years, then include these ages as knots;
2. Setting knots at scheduled visits is a sensible strategy for obtaining predictions at precisely the scheduled times. Set equidistant knots if the analysis requires a fixed time interval;
3. Keep the number of knots low, for speed and simplicity. Having many (≥ 10) knots can improve the fit to the data. Still, it will also increase calculation time and may result in unstable solutions. For problems that require more than 10 knots, reduce calculation time by the setting `method = "kr"` method, and use `control_kr()` to improve stability by a correlation model;
4. Do not place knots in sparsely filled areas of the data, e.g. in-between two visits. Doing so may result in erratic joins;
5. Define a starting time common to all subjects (e.g. birth) and set the first breakpoint `knots[1]` equal to the left boundary knot `boundary[1]`. The `brokenstick()` is already cautions to ensure this;
6. Order knots in size;
7. Use the `get_knots()` function to extract knots from a fitted model;
8. Set maximum value in `knots` to the highest time of scientific interest, but still within the data range. Set boundary knot `boundary[2]` larger than this value, e.g. equal to the maximum of the time variable. Broken stick estimates at the right boundary knot have no useful interpretation, so exclude those estimates from plots and ignore them in subsequent analyses;
9. Rule of thumb: Limit the number of knots to the (average) number of data points per subject;
10. Set knots to explicit values to support generalisation over the time variable in the training data.

5. Applications

5.1. Critical periods

The following question motivated the development of the broken stick model: *At what ages do children become overweight?* Knowing the answer to this question provides handles for preventive interventions to counter obesity. Dietz (1994) suggested the existence of three critical periods for obesity at adult age: the prenatal period, the period of adiposity rebound (roughly around the age of 5-6 years), and adolescence. Obesity formed in these periods is likely to increase the obesity risk at adult age and its complications.

A growth period, bounded by ages T_1 and T_2, is critical for adult overweight if the following criteria hold: (de Kroon *et al.* 2010)

a. there is a significant difference in mean gain score $Z_2 - Z_1$ between subjects with and without adult overweight;

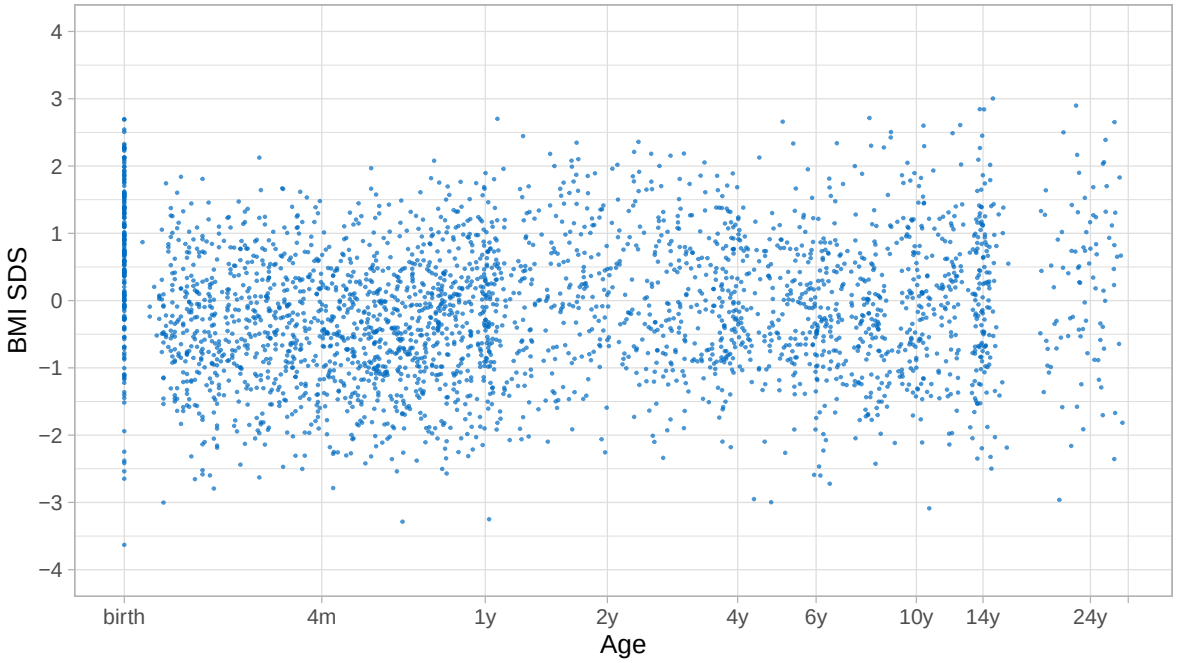

Figure 10: Body Mass Index (BMI) SDS by log(age + 0.2) (Terneuzen cohort)

b. the gain score $Z_2 - Z_1$ has an independent contribution over Z_2 to the prediction of Z_{adult}. It not only matters where you were at T_2 but also how you got there;
c. Z_2 correlates highly with Z_{adult}, so it is easier (i.e. with higher sensitivity and specificity) to identify children at risk for adult overweight.

de Kroon *et al.* (2010) found that the age interval 2-6 years met all criteria for a critical period. Our re-analysis tests the requirements for the following age intervals: birth-4 months, 4 months-1 year, 1-2 years, 2-4 years, 4-6 years, 6-10 years and 10-14 years. Hence, we define the following break ages:

```
R> knots <- round(c(0, 1/3, 1, 2, 4, 6, 10, 14, 24, 29), 3)
R> labels <- c("birth", "4m", "1y", "2y", "4y", "6y", "10y", "14y", "24y", "")
```

The Terneuzen Birth Cohort (de Kroon, Renders, Kuipers, van Wouwe, van Buuren, de Jonge, and Hirasing 2008) comprises of 2604 children born around the year 1980 in Terneuzen, The Netherlands. Figure 10 shows the BMI standard deviation scores (SDS) against age in a random subset of 306 children. While we may easily recognise scheduled visits at birth, 1y and 14y, observations at other periods are less structured. Compared to the analysis in de Kroon *et al.* (2010), we removed the knots at 8 days and 18 years (because these appear in sparse data areas) and added knots at 4, 14 and 24 years. We set the right boundary knot to 29y, slightly higher than the maximum age in the data.

```
R> ctl <- control_brokenstick(
+   lmer = lmerControl(check.conv.grad = .makeCC("warning", 0.02, NULL),
+                      check.conv.singular = .makeCC("ignore", 0.001)))
```

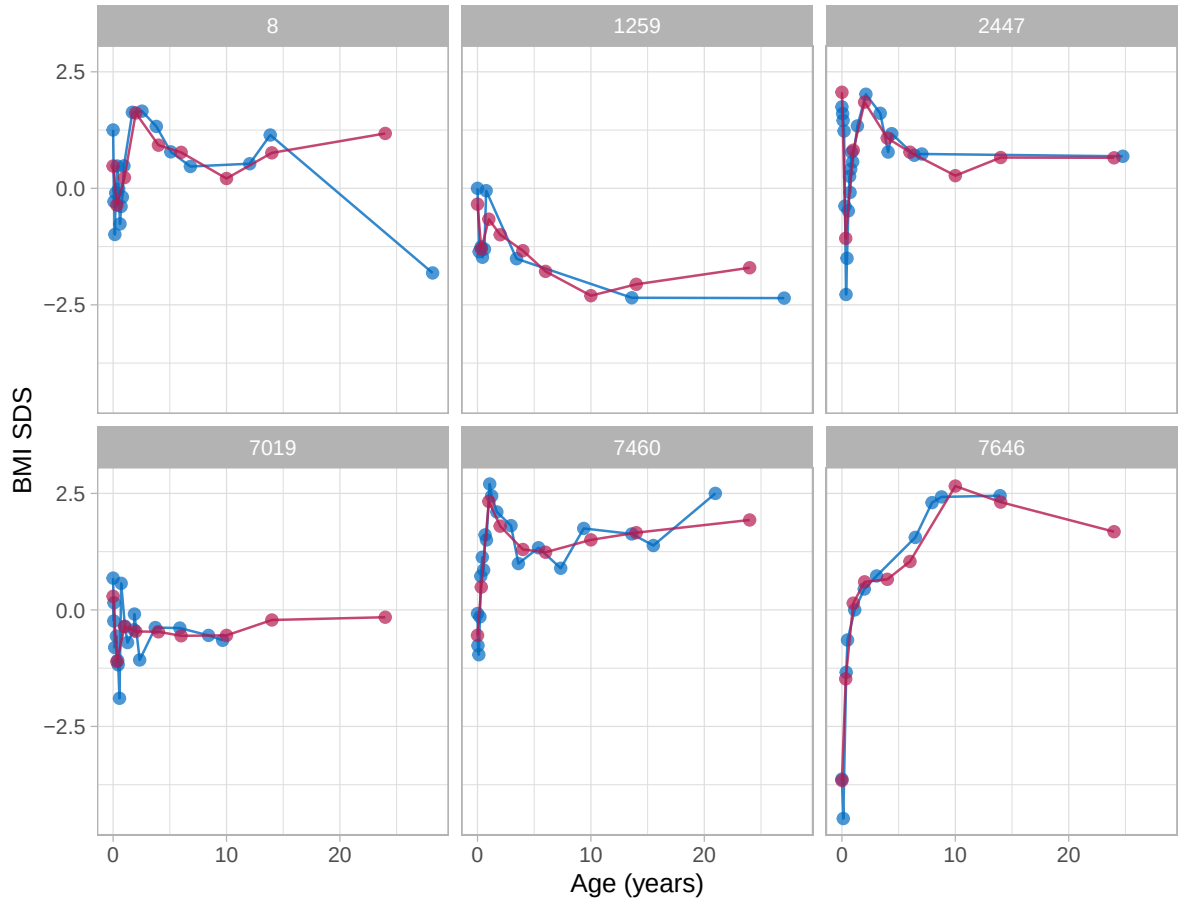

Figure 11: Body Mass Index (BMI) SDS trajectories of six subjects, observed (blue) and fitted (red). lmer method.

```
R> fit_lmer <- brokenstick(bmi.z ~ age | id, data = mice::tbc,
+                          knots = knots, boundary = c(0, 29),
+                          control = ctl)
```

Depending on the precise specification of the knots, the default `brokenstick()` procedure that calls `lme4::lmer()` may print the warning `Model failed to converge with max|grad| = 0.011882 (tol = 0.002, component 1)` or a message `boundary (singular) fit: see ?isSingular`. These issues arise because of the over-parametrised nature of the default broken stick model, potentially resulting in a singular variance-covariance matrix `fit@omega`, combined with a sparsity of data. The `control_brokenstick()` command can prevent the warning and message.

```
R> ids <- c(8, 1259, 2447, 7019, 7460, 7646)
R> plot(fit_lmer, mice::tbc, group = ids,
+      ylab = "BMI SDS", xlab = "Age (years)")
```

Figure 11 shows observed and fitted BMI SDS trajectories for six subjects using the `lmer` method. In general, the model fits the data well. The per cent explained variance of BMI

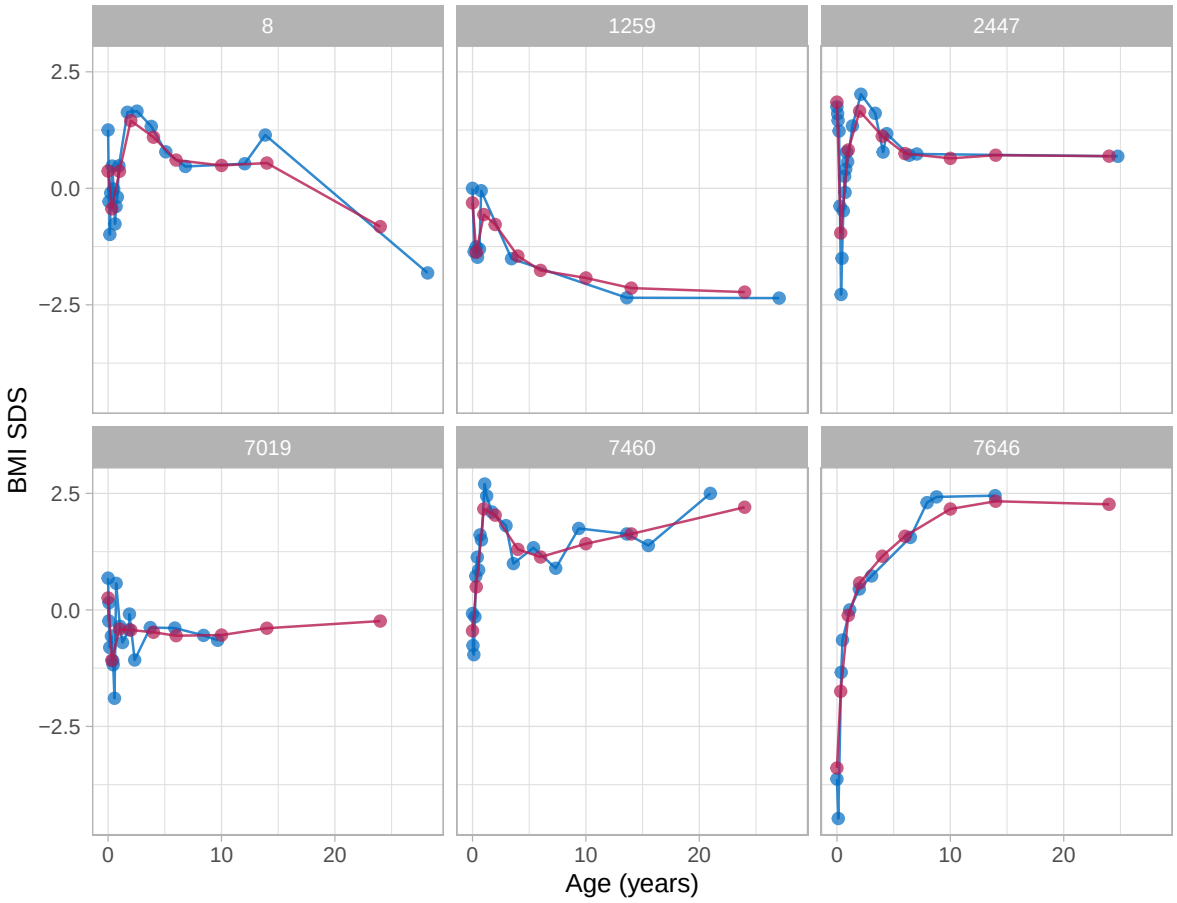

Figure 12: Body Mass Index (BMI) SDS trajectories of six subjects, observed (blue) and fitted (red). kr method.

SDS obtained by `get_r2(fit_lmer, mice::tbc)` equals 84 per cent. Note that he fitted trajectory for subject 8 reveals a pretty rough estimate at the age of 24y. Persons 1259 and 7460 have very low (-2.5 SD) and high (+2.5 SD) BMI SDS at adult age, respectively. Note that the model pulls the adult BMI SDS estimates (in red) towards the global mean, due to the well-known bias-variance tradeoff.(Gelman and Hill 2007, pp. 394) Pulling is more vigorous at the extremes. The effect is negligible for more average trajectories, such as for subject 2447.

The royal way to treat such warnings and message is to simplify the model, e.g., by removing knots. An alternative is to constrain the broken stick model, in particular the covariance matrix. Selecting the `kr` method applies the Argyle correlation model, at the expense of fit to the data. On the other hand, the fitted trajectories will be much stabler in regions with sparse data. The following snippet applies the `kr` method.

```
R> fit_kr <- brokenstick(bmi.z ~ age | id, data = mice::tbc,
+                        knots = knots, boundary = c(0, 29),
+                        method = "kr", seed = 41441)
```

Figure 12 is the equivalent to Figure 11, but now for method `kr`. The per cent explained

variance is the same. Due to the constraint placed on the covariance matrix, the trajectories are slightly smoother and more stable in the adult ages with limited data. The rough estimate for subject 8 has gone. There is still some gravity towards to global mean at knot 24y for persons 1259 and 7460, but it is of lesser magnitude. All fitted trajectories are well behaved. We, therefore, select the `kr` solution for further analysis.

To identify critical periods, we need to predict adult overweight. In Figure 12, only three out of six subjects had a BMI measurement at adult age. Since we do not want the results to overly depend on fitted extrapolations, we restrict the analysis sample to persons with an adult measurement. The following lines extract the repeated measures for 92 (out of 306) individuals for whom we observed adult BMI.

```
R> tbc1 <- mice::tbc %>%
+   filter(!is.na(ao) & first) %>%
+   select(id, nocc, sex)
R> tbc2 <- mice::tbc.target %>%
+   filter(id %in% tbc1$id)
R> prd <- predict(fit_kr, mice::tbc, x = "knots",
+                 shape = "wide", group = tbc1$id)
R> data <- bind_cols(prd,
+                    select(tbc1, -id),
+                    select(tbc2, -id))
R> head(data, 3)
```

```
# A tibble: 3 x 15
     id   `0` `0.333`     `1`    `2`    `4`    `6`   `10`   `14`   `24`   `29`
  <dbl> <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1     8 0.371  -0.440  0.366   1.45   1.10   0.606  0.490  0.543 -0.822 -1.18
2    60 0.159  -0.372 -0.0441 -0.361 -0.595 -0.819 -1.03  -1.16  -1.46  -1.40
3    97 1.68    0.569  0.948   1.90   1.28   0.838  0.444  0.238  0.398  0.709
# ... with 4 more variables: nocc <dbl>, sex <dbl>, ao <dbl>, bmi.z.jv <dbl>
```

Figure 13 shows the 92 fitted trajectories coloured by adult overweight status (BMI SDS > 1.3). It is evident that BMI SDS at ages of 14y or 10y is highly predictive of adult overweight, but does that also hold in early childhood? Also, does a change in specific periods predict later overweight? To answer such questions, we fit simple linear models to predict observed (not fitted!) BMI SDS at adult age from the fitted BMI SDS trajectories. The following code block fits two models for the period 4y-6y.

```
R> m1 <- lm(bmi.z.jv ~ `6`, data)
R> m2 <- lm(bmi.z.jv ~ `6` + I(`6`-`4`), data)
R> anova(m1, m2)
```

```
Analysis of Variance Table

Model 1: bmi.z.jv ~ `6`
Model 2: bmi.z.jv ~ `6` + I(`6` - `4`)
```

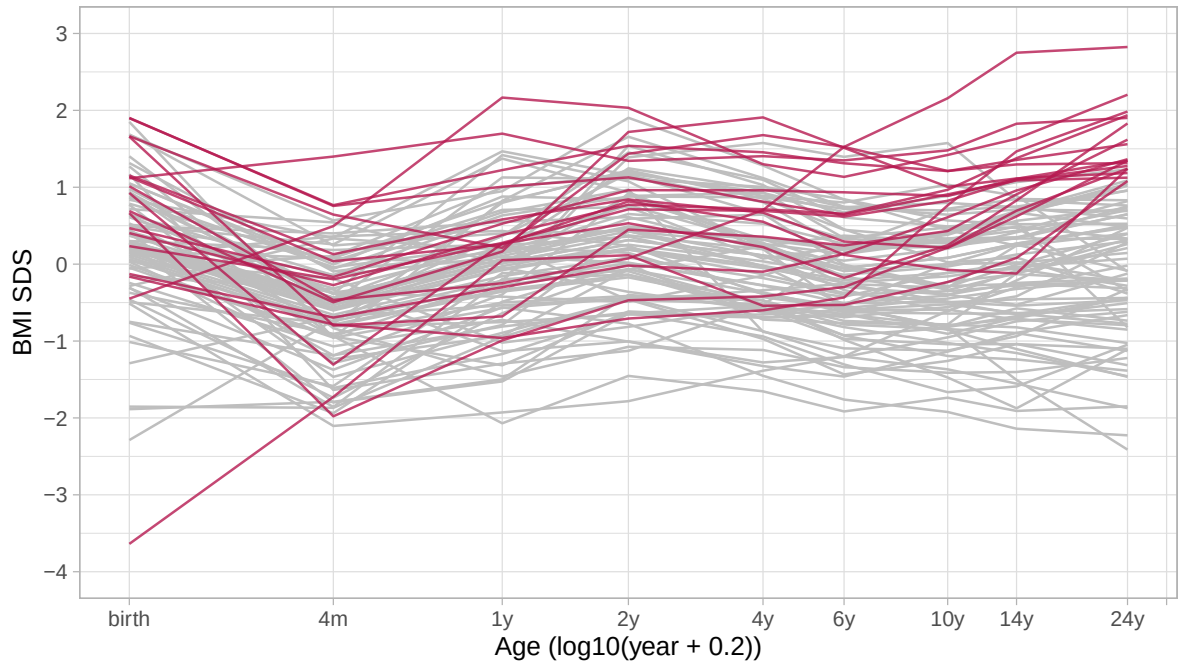

Figure 13: Body Mass Index (BMI) SDS trajectories for 92 subjects, coloured by adult overweight status.

```
  Res.Df  RSS Df Sum of Sq    F  Pr(>F)
1     90 74.3
2     89 63.5  1      10.8 15.2 0.00019 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model `m1` predicts adult BMI SDS from BMI SDS 6y, and explains 45.3 per cent of the variance. Model `m2` extends the model with the pre-gain between 4y and 6y. If the pre-gain improves the prediction, then it matters how much you gained between 4y and 6y. In that case, we would call the interval 4y-6y a critical period. Here we found that model 2 explain 53.6 per cent variance, thus 8.3 per cent more. The `anova` statement performs the formal test. In this case, the pre-gain is significant over the last predictor at 6y. Thus, interval 4y-6y classifies as a critical period. We can repeat these analyses for other age intervals, similar to Table 3 in Kenward (1987).

5.2. Time-to-time correlations

The *conditional gain score* is defined as (Cole 1995)

$$\text{conditional } Z_{\text{gain}} = \frac{Z_2 - rZ_1}{\sqrt{1-r^2}},$$

where Z_1 and Z_2 are the standard deviation scores at times T_1 and T_2, with $T_2 > T_1$, and where r is the correlation between Z_1 and Z_2. The conditional gain corrects for regression to the mean, which is its selling point over traditional velocity measures and is less sensitive to

measurement error.(van Buuren 2007) A practical difficulty is to obtain r for a given T_1 and T_2. The time-to-time correlation matrix needs to be known. Also, we need to interpolate r if T_1 or T_2 differs from the tabulated ages.

The broken stick model provides an estimate of the time-to-time correlation matrix. The `brokenstick` object stores the variance-covariance matrix Ω of the random effects in the list component `fit$omega`. For a perfectly fitting model (with $\sigma^2 = 0$) Ω equals the time-to-time covariance matrix, so `cov2cor(fit$omega)` gives the desired time-to-time correlation matrix. If $\sigma^2 > 0$ then Ω overestimates the covariances between the observed data. In general, we need to add the within-residual variance estimate to the diagonal, thus $\Omega + \hat{\sigma}^2 I(n_i)$ to estimate the time-to-time covariance matrix.

```
R> fit <- brokenstick(hgt.z ~ age | id, data = smocc_200,
+                     knots = 1:4/2, boundary = c(0, 3))

boundary (singular) fit: see ?isSingular

R> t2t <- fit$omega + diag(fit$sigma2, ncol(fit$omega))
R> round(cov2cor(t2t), 2)

        age_0 age_0.5 age_1 age_1.5 age_2 age_3
age_0    1.00    0.52  0.41    0.42  0.35  0.01
age_0.5  0.52    1.00  0.77    0.73  0.68 -0.30
age_1    0.41    0.77  1.00    0.81  0.79 -0.16
age_1.5  0.42    0.73  0.81    1.00  0.84  0.00
age_2    0.35    0.68  0.79    0.84  1.00  0.07
age_3    0.01   -0.30 -0.16    0.00  0.07  1.00
```

In child growth, we expect that the correlation tapers off as the difference between T_1 and T_2 grows. Also, for a fixed interval $T_2 - T_1$ we expect the correlation to increase with age. Ignoring the uninteresting estimate for `age_3`, we find that both expectations hold. Altering the number and location of the knots may change this. It is often useful to scan the time-to-time correlation matrix for gross deviations of the expectations. If such happens, one could simplify the model, for example, by subjecting Ω to a correlation model.

```
R> fit <- brokenstick(hgt.z ~ age | id, data = smocc_200,
+                     knots = seq(0, 2, 0.1), boundary = c(0, 3),
+                     method = "kr")
R> t2t <- fit$omega + diag(fit$sigma2, ncol(fit$omega))
R> dim(t2t)

[1] 22 22
```

The above code fits a model with 22 equidistant breakpoints, which is likely large enough for most purposes. It works because it restricts the covariance-matrix by the Argyle correlation model, which summarises the information by just two parameters. We may extract or re-estimate these parameters and create a one-liner for calculating r.

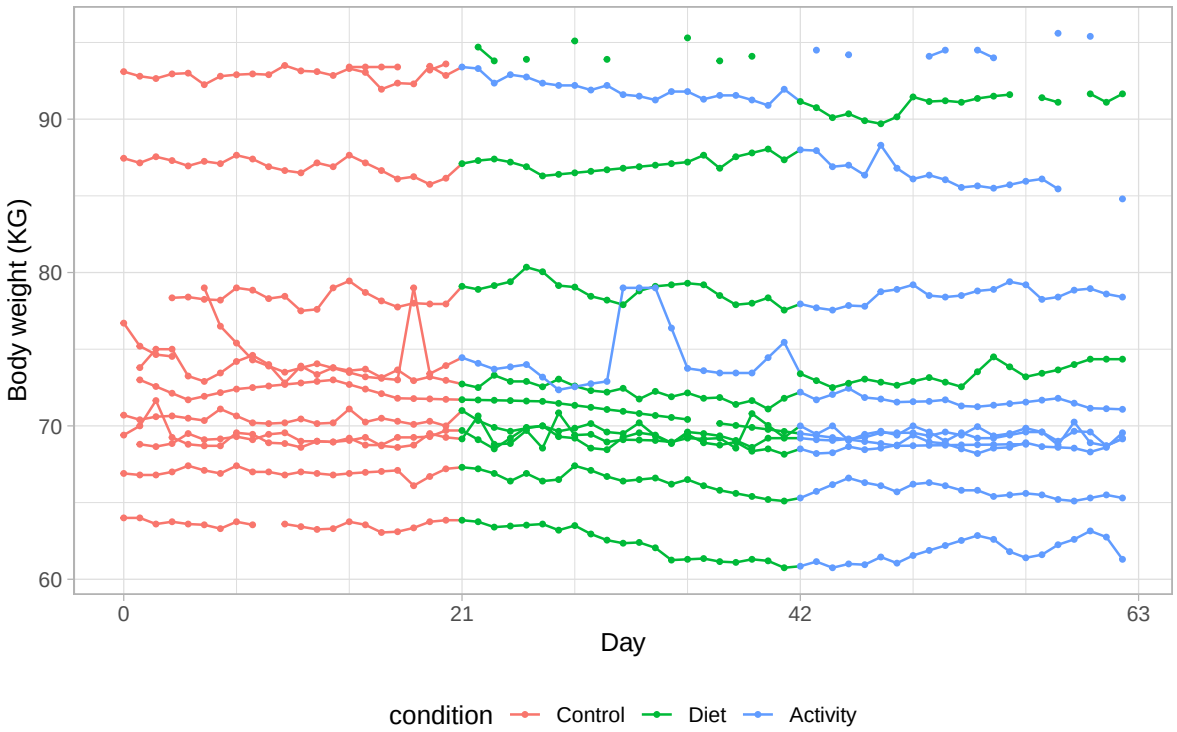

Figure 14: Daily body weight (KG) for 12 subjects under three conditions.

We cannot indefinitely add breakpoints. Suppose we double the number of knots by setting `knots = seq(0, 2, 0.05)`. Then even `kr` is not able to cope and will abort with `Error: Sigma is symmetric but not positive`. Thus, as always, be sensible in what you ask the software to do for you.

5.3. Profile analysis

Profile analysis (Morrison 1976; Johnson and Wichern 1988) refers linear multivariate linear methods to test for differences in population means or treatment effects, typically by regression analysis or multivariate analysis of variance (MANOVA). These methods assume independence of subjects, organise the data at the subject level, and express parameters of interest by linear combinations of outcomes, like change scores, means or other derived quantities.

Krone, Boessen, Bijlsma, van Stokkum, Clabbers, and Pasman (2020) report a statistical analyses using the linear mixed model with time-varying individual subject data. This section re-analyses the data from Figure 4 using the broken stick model. The data are available as the `brokenstick::weightloss` object.

Figure 14 charts daily body weight measurements of twelve individuals who were followed for nine weeks. The investigators subdivided the total duration into three periods of three weeks. Period one (week 1-3) acted as a control period. During period 2 (week 4-6), the investigators stimulated participants to restrict food intake, and during period 3 (week 7-9) the experimenters promoted physical activity. Subjects 4 and 12 received the interventions in the opposite order. See Krone *et al.* (2020) for more detail.

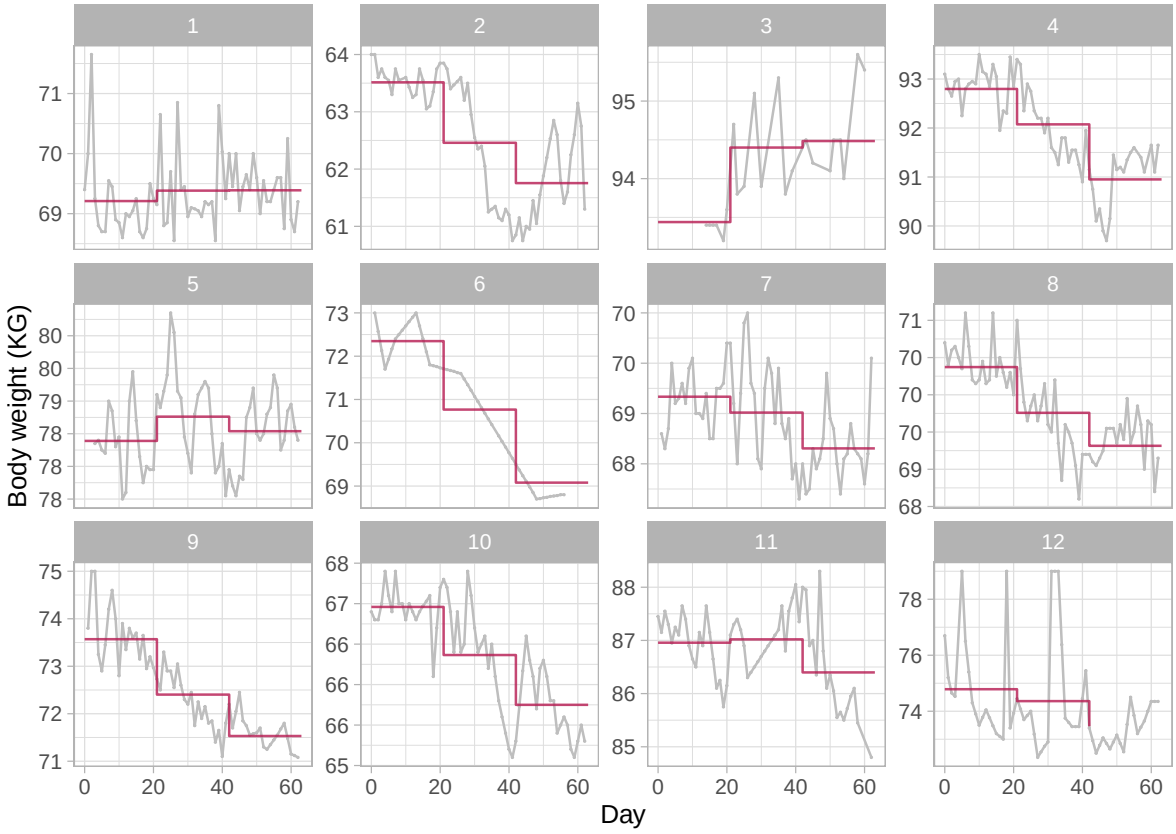

Figure 15: Constant model. Observed and fitted trajectories for a model that summarises each experimental period by a constant.

Most of these subjects adhere quite well to the data collection design. Some trajectories show gaps due to missed measurements. The most extreme example is the trajectory at the top, which has only scant measures. Other curves display stretches of lines, suggesting that missed measurements were linearly interpolated. One of the series shows some surprising spikes, likely to be measurement errors. All in all, these data perfectly illustrate the inescapable imperfections of real data.

The remainder of the section discusses two ways to estimate the effect of diet and physical activity on body weight.

Constant model

```
R> fit0 <- brokenstick(body_weight ~ day | subject, data,
+                      knots = c(0, 21, 42, 63), degree = 0)
R> plot(fit0, data, size_y = 0, color_y = rep("grey", 2), what = "all",
+      scales = "free_y", xlab = "Day", ylab = "Body weight (KG)",
+      n_plot = 12, ncol = 4)
```

The model underlying Figure 15 summarises the trajectory within a period by a constant, the mean. We obtain an estimate of these mean by setting the `degree = 0` argument. This

model gives a fair representation of the trajectory of subjects 9 (a persistent downward trend), 1 and 5 (no trend). On the other hand, the model fails to capture patterns for subjects 2, 4 and 8 (rebound in period 3) or 11 (inverse rebound).

It is straightforward quantify the effects of `Diet` and `Activity` relative to `Control`. The next code snippet calculates these effects per person, accounting for the intervention order reversal for subjects 4 and 12.

```
R> prd <- data.frame(predict(fit0, data, x = "knots", shape = "wide"))
R> control <- prd[, 2]
R> diet <- prd[, 3]
R> diet[c(4, 12)] <- prd[c(4, 12), 4]
R> activity <- prd[, 4]
R> activity[c(4, 12)] <- prd[c(4, 12), 3]
R> round(data.frame(diet_control = diet - control,
+                   activity_control = activity - control,
+                   activity_diet = activity - diet), 1)
```

```
   diet_control activity_control activity_diet
1           0.2              0.2           0.0
2          -1.1             -1.8          -0.7
3           1.0              1.0           0.1
4          -1.8             -0.7           1.1
5           0.4              0.1          -0.2
6          -1.6             -3.3          -1.7
7          -0.2             -0.5          -0.4
8          -0.6             -1.1          -0.4
9          -1.2             -2.0          -0.9
10         -0.6             -1.2          -0.6
11          0.1             -0.6          -0.6
12         -1.3             -0.4           0.9
```

The average weight under caloric restriction is 0.6 KG lower than control. In contrast, we find a 0.8 KG lower body weight when we stimulate physical activity. We could be tempted to believe that exercise reduces weight more than a diet. However, except for subjects 4 and 12, the investigators administered the activity treatment after the diet treatment, so the difference relative to control represents the *combined effect* of diet and activity on body weight. It might be more relevant to study the difference between training and diet (third column). The average difference of -0.3 KG suggests that diet is more effective than physical activity. Realise that also this estimate is not entirely satisfactory. First, subjects 4 and 12 had a reversed administration, so the difference does not make sense for them. Second, as anyone who has tried to lose weight can attest, "quick wins" are more likely in period 2 than in period 3. Although it is possible to account for these sequence effects, there is a more intuitive analysis of the data.

Broken stick model

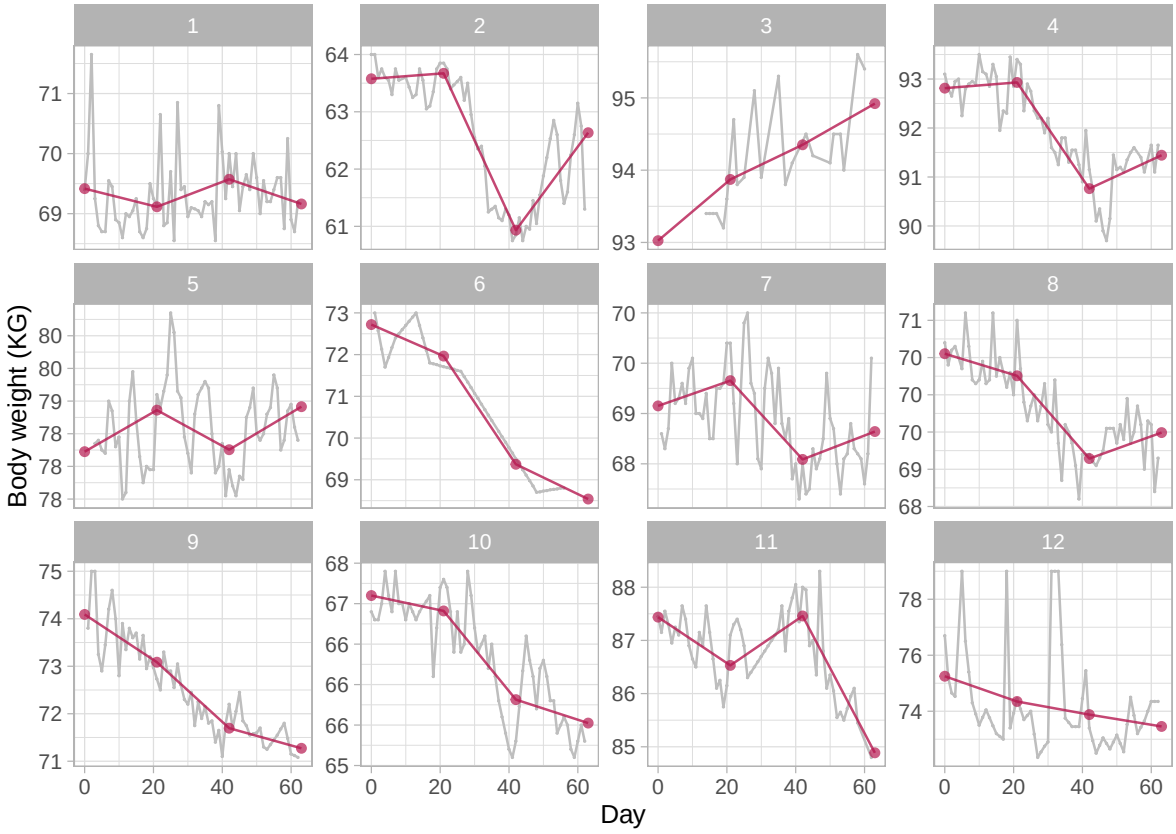

Figure 16: Broken stick model. Observed and fitted trajectories for a model that summarises each experimental period by a line.

```
R> fit1 <- brokenstick(body_weight ~ day | subject, data,
+                      knots = c(0, 21, 42, 63))
R> plot(fit1, data, size_y = 0, color_y = rep("grey", 2), what = "all",
+      size_yhat = 1.5, scales = "free_y", , xlab = "Day", ylab = "Body weight (KG)",
+      n_plot = 12, ncol = 4)
```

Figure 16 shows the same data as in Figure 15 but now fitted by the broken stick model. This model also suggests a persistent downward trend for subject 9 and an absence of for participants 1 and 5. Also, the model now correctly identifies the prominent zig-zag patterns for persons 2, 4, 8 and 11 across the three experimental periods.

A natural way to quantify the effect of the intervention is to calculate the before-after estimate per period. For example, for person 2 the effect of diet is $60.9 - 63.6 = -2.7$ KG, of activity is $62.6 - 60.9 = +1.7$ KG. The following code accounts for the alternate treatment ordering of subjects 4 and 12.

```
R> prd <- data.frame(predict(fit1, data, x = "knots", shape = "wide"))
R> control <- prd[, 3] - prd[, 2]
R> diet <- prd[, 4] - prd[, 3]
R> diet[c(4, 12)] <- prd[c(4, 12), 5] - prd[c(4, 12), 4]
```

```
R> activity <- prd[, 5] - prd[, 4]
R> activity[c(4, 12)] <- prd[c(4, 12), 4] - prd[c(4, 12), 3]
R> round(data.frame(control = control,
+                   diet = diet,
+                   activity = activity), 1)

   control diet activity
1     -0.3  0.5     -0.4
2      0.1 -2.7      1.7
3      0.8  0.5      0.6
4      0.1  0.7     -2.2
5      0.6 -0.6      0.7
6     -0.8 -2.6     -0.8
7      0.3 -0.8      0.3
8     -0.3 -1.1      0.3
9     -1.0 -1.4     -0.4
10    -0.2 -1.1     -0.3
11    -0.9  0.9     -2.6
12    -0.9 -0.4     -0.5
```

The average effects are -0.2 KG (control), -0.7 KG (diet) and -0.3 KG (activity). Although not statistically significant, the slight decrease of -0.2 KG during the control period suggests that weight monitoring by itself may motivate the participant to lose weight. The effect estimates for diet and activity are of similar magnitude as before. Still, they can be sizeable discrepancies at the individual level, e.g. for subjects 2 or 11.

We may obtain a simple estimate of the sequence effect by linear regression as

```
R> df <- data.frame(y = c(diet, activity),
+                   act = rep(c(0, 1), each = 12),
+                   per2 = rep(c(rep(1, 3), 0, rep(1, 7), 0), 2))
R> coef(lm(y ~ act, data = df))

(Intercept)         act
      -0.68        0.38

R> coef(lm(y ~ act + per2, data = df))

(Intercept)         act        per2
      -0.79        0.38        0.12
```

The result is a little surprising. When applied in period 2, the intervention leads to higher body weight (+123 grammes) as compared to administration in period 3. Of course, bear in mind that we calculated these results on very few individuals. Hence, they are sensitive to substantial estimation error.

This application demonstrates that the broken stick model can effectively capture rapid linear changes in experiments. Even though the actual timing of the observations may be erratic,

it is easy to define, interpret and calculate intuitive effect estimates at the individual level. Note that the analysis here assumed an instantaneous effect of the interventions. If we expect a delay, then we may right-shift the knots by a few days and re-estimate the broken stick model. By varying the number of days, we may be able to detect the optimal delay factor.

5.4. Curve interpolation

Problem

A growth chart visualises the individual trajectory relative to a set of centile lines. We may store a centile line as a set of coordinates with a relatively dense age grid. If we connect the adjacent vertices by a straight line, the centile will appear as smooth in time. However, this plotting method runs into trouble when ages are wide apart. This section shows how we can create a realistic interpolation with sparse time data.

Interpolation in measurement scale

Suppose we measured the length of a boy at the ages of 1 month (52.6 cm) and 14 months (81.7 cm). The following code block uses the `AGD::y2z()` function to convert the measurements to standard deviation scores (SDS) relative to the reference of the Fourth Dutch Growth Study.

```
R> boy <- data.frame(x = c(1/12, 14/12), y = c(52.6, 81.7))
R> ref <- AGD::nl4.hgt
R> boy$z <- AGD::y2z(y = boy$y, x = boy$x, sex = "M", ref = ref)
R> boy$z

[1] -0.98  0.99
```

During the period the boy grows from moderately short (about -1.0 SD at month 1) to relatively tall (about +1.0 SD at month 14). Figure 17 shows the usual representation of the growth chart with a straight line drawn between the two values. Due to the convex shape of the centile lines, the straight line that connects the two measurements starts at -1.0 SD, then touches the -2.0 SD centile around 0.3 yr, is back at -1.0 SD around 0.7 yr, crosses the 0.0 SD line at 1 yr, and ends at +1.0 SD at 1.2 yr. The graph on the right-hand side portrays the interpolated growth curve in the Z-score scale. Since length growth during infancy is not linear in time, finding a real growth curve like this is extremely unlikely. Since we have just two data points smoothing the data does not help either.

Interpolation in the Z-score scale

A first alternative is to apply the linear interpolation in the Z-score scale. This option is attractive because convexity of centile lines is absent on this scale.

Figure 18 illustrates the interpolation in the Z-scale. By definition, the line that connects the measurements is straight in the Z-score scale. In the cm scale, the representation is more realistic and more pleasing to the eye. The curve crosses the 0 SD line about halfway, at about 0.6 yr.

While this approach is a considerable improvement over interpolation in the Y-scale, it is still not ideal. The assumption underlying this interpolation is that the Z-score increment is

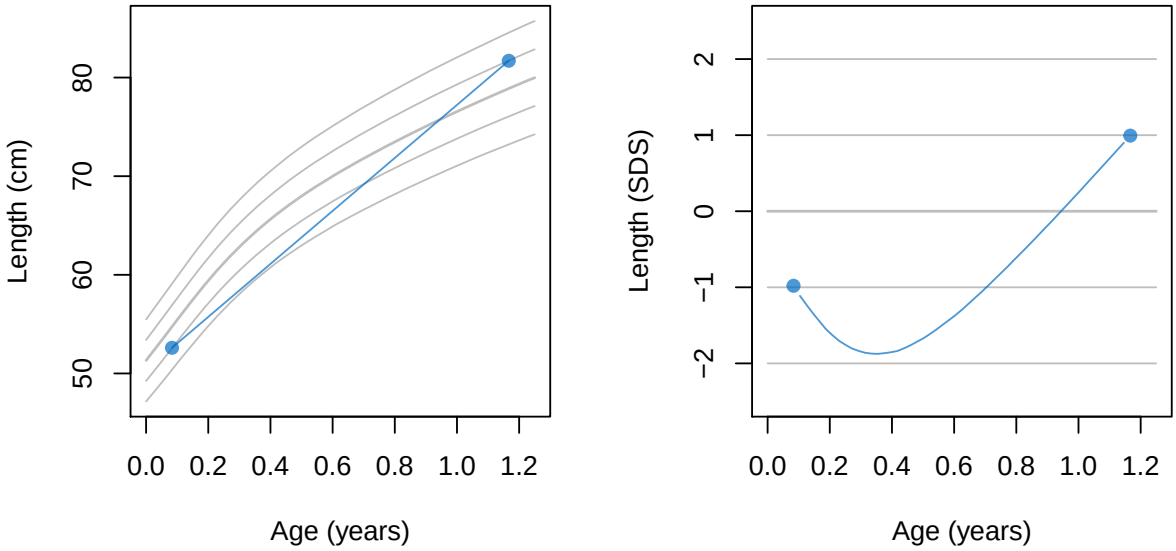

Figure 17: Linear interpolation in the cm scale results in an unrealistic trajectory at intermediate ages.

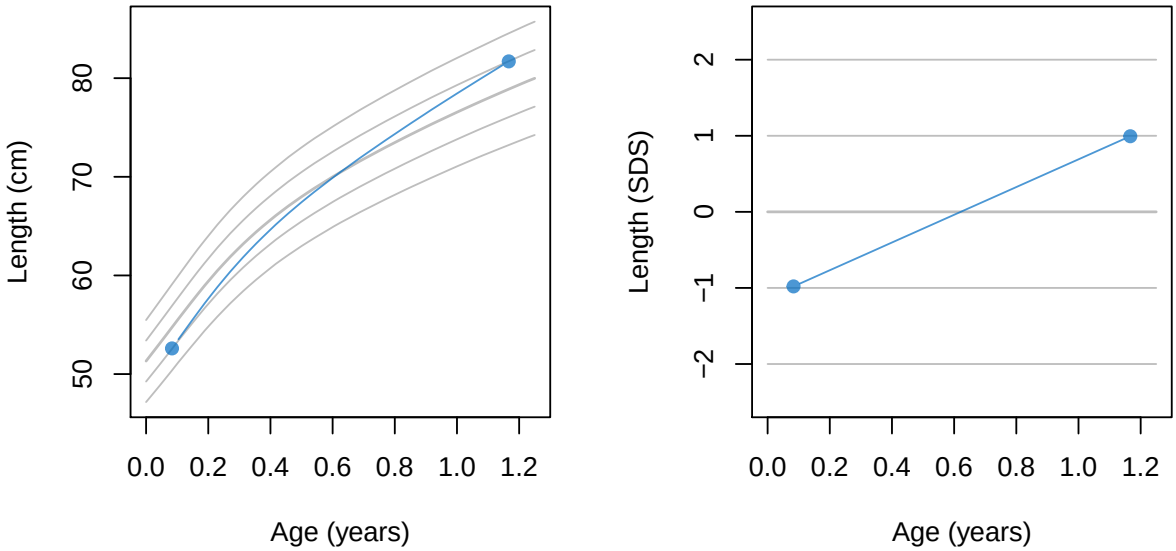

Figure 18: Linear interpolation in the Z-score scale results in a more realistic trajectory at intermediate ages.

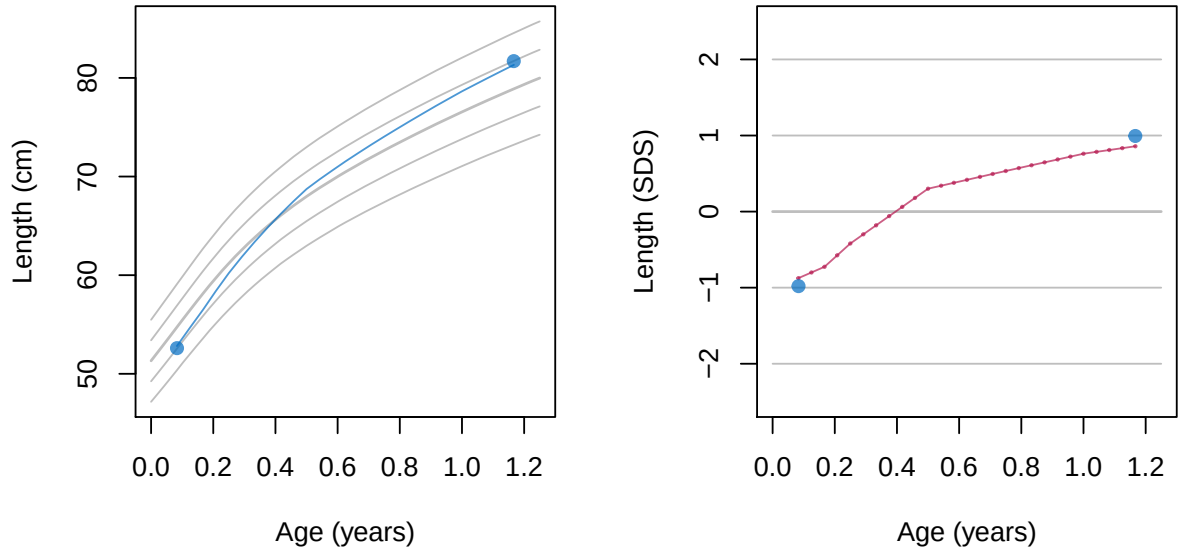

Figure 19: Broken stick model fitted in the SDS scale results in a most realistic expected trajectory at intermediate ages.

constant across time. This assumption is false, however. Since length growth is more variable during the first half-year than in the second half-year, we expect that the larger share of the increment to occur during the earlier months. In other words, the cross-over point at 0.6 yr is too late.

Interpolation by the broken stick model

The second alternative is a model-based interpolation. Assuming the availability of a fitted broken stick model, we specify a dense time grid, say every week, and predict the length at these times given the data from the observed trajectory. The expected curve represents the most likely values under the model at the intermediate ages. The following code calculates the relevant estimates from the `fit_200` fitted model:

```
R> # prepare data input
R> age <- round(seq(2/24, 28/24, 1/24), 3)
R> z <- rep(NA, length(age))
R> z[1] <- boy$z[1]; z[length(z)] <- boy$z[2]
R>
R> # predict with broken stick model
R> zout <- predict(fit_200, x = age, y = z, shape = "vector")
R>
R> # convert predicted values to Y-scale
R> yout <- z2y(x = age, z = zout, ref = ref)
```

Figure 19 shows the results of the broken stick model. The predicted curve in the Y-scale represents the most likely course according to the broken stick model. Because growth is more variable during early infancy, the child realises the larger share of the change during the first part of the period. As a result, the cross-over point where the predicted value intersects the

0 SD line is now at 0.4 yr, considerably earlier than obtained by the two other interpolation methods. The plot on the right-hand side confirms the steeper slope in the first part. Note that this method treats rising and declining curves alike. For example, if the boy's length would be 57 cm at month 1 (+1.0 SD) and 76 cm at month 14 (-1.0 SD), the cross-over point would also be at 0.4 yr.

Observe that we left the world of pure interpolation and moved to an approximation of the data by a model. The observed and predicted lengths are not exactly equal. The difference is so small that we may hardly notice the discrepancy when plotted in the Y-scale, but it is more conspicuous in the Z-score scale. Of the three approaches considered here, the broken stick model provides the most realistic expected trajectory at the intermediate ages.

5.5. Multiple imputation

Remember from section 2 that the broken stick estimates are conditional means. We may be tempted to analyse these estimates as if they were "just data", but they do not have the same variability as the real data. For example, suppose we calculate the correlation matrix of the broken stick estimates. We know that the values in this matrix will exceed those from the underlying observed data. Not accounting for this fact leads to overconfident predictions and results that are too good to be true.

Multiple imputation (Rubin 1987; van Buuren 2018) restores variability by adding noise. We may fit standard complete-data software to the imputed data, and obtain valid regression weights, confidence intervals and P-values under a wide range of conditions.

By default, method `kr` executes 200 iterations of the Kasim-Raudenbush sampler. The `imp_skip` argument to the `control_kr()` function specifies the interval at which the method adds noise to the broken stick estimates. The following code block appends the break ages to the input data and sets `imp_skip = 10`. The call to the `brokenstick()` function thus creates 20 imputations for each missing outcome (`hgt.z` here).

```
R> ctl <- control_brokenstick(kr = control_kr(imp_skip = 10))
R> knots <- round(c(0, 1, 2, 3, 6, 9, 12, 15, 18, 24)/12, 4)
R> data <- bind_rows(smocc_200[!is.na(smocc_200$hgt.z), ],
+                    expand.grid(id = unique(smocc_200$id), age = knots))
R> fit_kr <- brokenstick(hgt.z ~ age | id, data = data,
+                        knots = knots, boundary = c(0, 3),
+                        method = "kr", control = ctl, seed = 15244)
R> plot(fit_kr, new_data = data, show = c(TRUE, FALSE, TRUE),
+      group = c(10001, 10005, 10022),
+      xlab = "Age (years)", ylab = "Length (SDS)")
```

Figure 20 displays the observed data from three persons plotted on top of 20 imputed trajectories. The within-person within-time average over the grey trajectories approximates to the broken stick estimate (not shown here). The observed curve in each panel occasionally strays towards the boundaries of the grey bundle. This behaviour is as expected and indicates that the blue curve performs like a grey curve.

Section 5.2 showed how we can estimate the time-to-time correlation matrix. An alternative way is to calculate it from the imputed data, as follows:

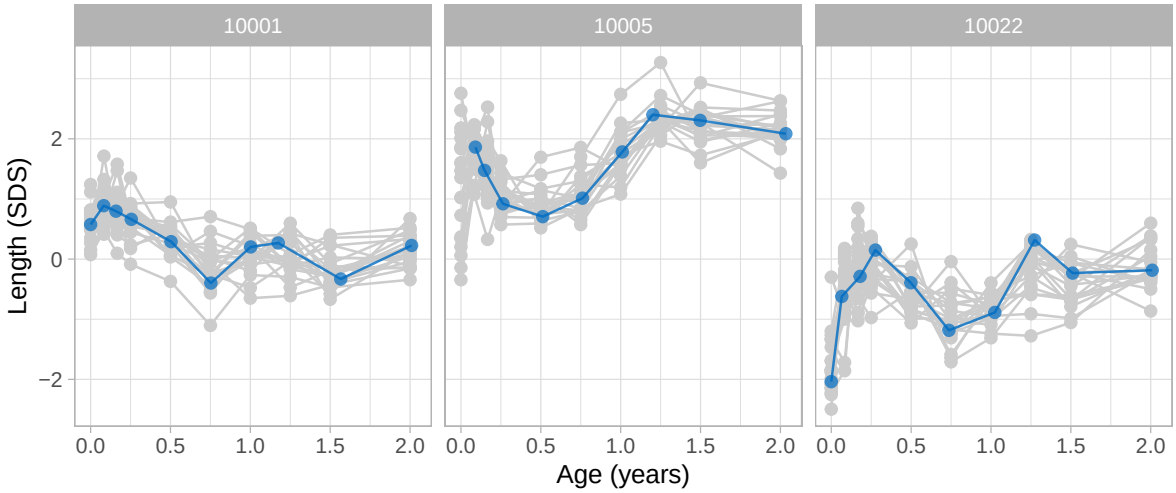

Figure 20: Observed data plotted on top of 20 imputed trajectories.

```
R> expand.grid(id = unique(smocc_200$id), age = knots) %>%
+   bind_cols(as.data.frame(fit_kr$draws)) %>%
+   pivot_longer(cols = starts_with("V"), names_to = "imp") %>%
+   pivot_wider(id_cols = c("id", "imp"), names_from = "age") %>%
+   select(-id, -imp) %>%
+   cor() %>%
+   round(2)
```

```
          0 0.0833 0.1667 0.25  0.5 0.75    1 1.25  1.5    2
0      1.00   0.68   0.62 0.56 0.46 0.39 0.37 0.35 0.33 0.29
0.0833 0.68   1.00   0.81 0.76 0.66 0.56 0.53 0.53 0.50 0.45
0.1667 0.62   0.81   1.00 0.84 0.72 0.63 0.58 0.57 0.56 0.51
0.25   0.56   0.76   0.84 1.00 0.77 0.68 0.63 0.61 0.59 0.54
0.5    0.46   0.66   0.72 0.77 1.00 0.84 0.80 0.76 0.73 0.67
0.75   0.39   0.56   0.63 0.68 0.84 1.00 0.86 0.81 0.78 0.72
1      0.37   0.53   0.58 0.63 0.80 0.86 1.00 0.86 0.83 0.77
1.25   0.35   0.53   0.57 0.61 0.76 0.81 0.86 1.00 0.88 0.81
1.5    0.33   0.50   0.56 0.59 0.73 0.78 0.83 0.88 1.00 0.84
2      0.29   0.45   0.51 0.54 0.67 0.72 0.77 0.81 0.84 1.00
```

Another important application of the multiply-imputed curves is to obtain correct confidence intervals and P-values for estimates of scientific interest. The most convenient way to do this is to convert the `brokenstick` object into an object of class `mids`, as defined by the **mice** package. The **brokenstick** package currently has no features that perform the conversion.

5.6. Curve matching

Curve matching (van Buuren 2014) is a tool to assist in the interpretation and prediction of individual growth curves. The idea is as follows. Suppose we measure the growth of the target child up to half a year and plot the measurements onto his or her growth chart. Curve

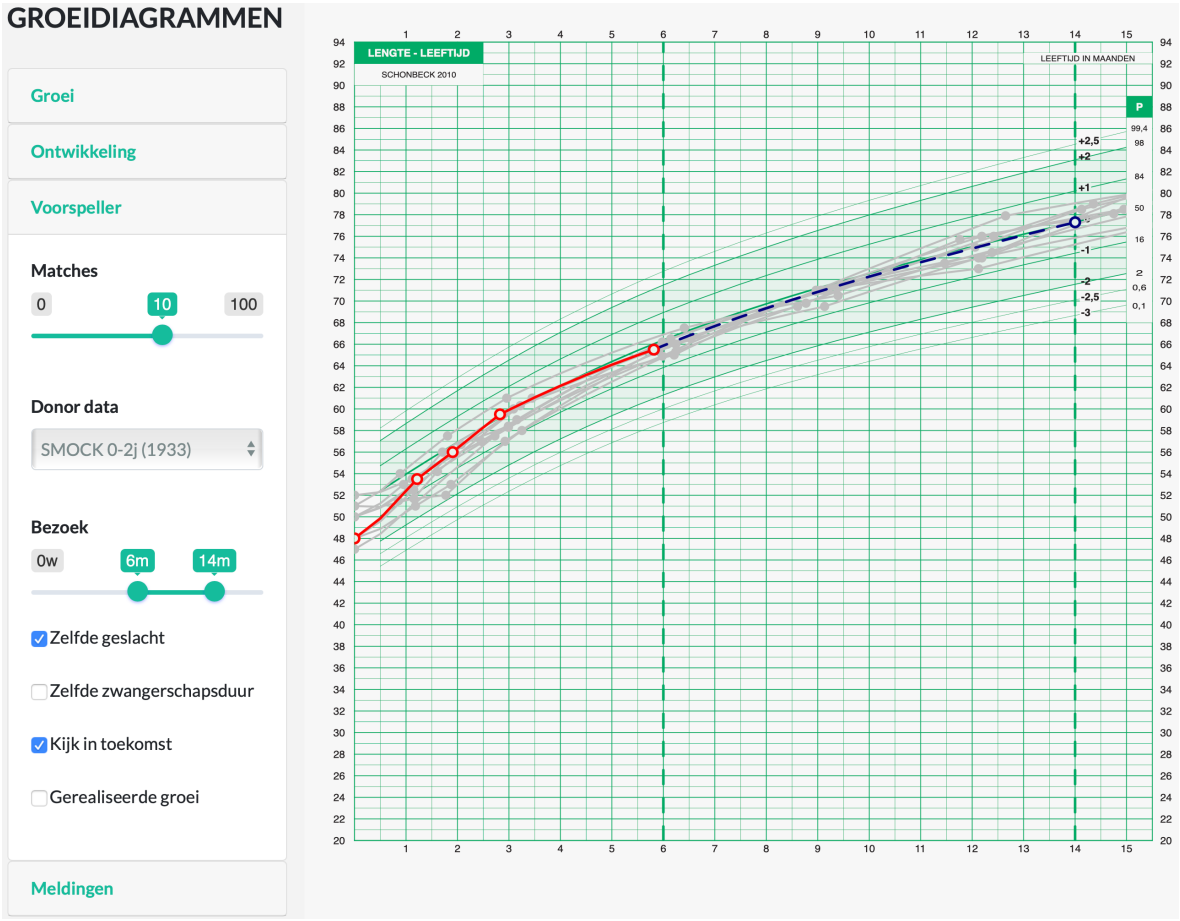

Figure 21: Curve matching. Predict infant length at 14 months given length data up to 6 months using 10 matches.

matching is a nearest-neighbour technique that relies on historical growth data. It finds, say, ten other children who are similar to the target child, and add the curves of those matches to the child's chart. If the matching is done right, then the bundle of historic growth curves suggests how the target child will develop in the future.

Figure 21 demonstrates curve matching for infant length. The red curve corresponds to five measurements of the target child made during the first six months. The ten grey curves are historic growth curves from the ten matched children. We may define similarity in many ways. Here we use a linear model to predict length at the age of 14m from previous length data. The distance between the target child and another child is equal to the difference between their predicted values. The procedure lifts the data of the matches from the database, and plots the observed growth curves onto the chart as grey curves. This method for finding nearest neighbours is known as *predictive mean matching* and has grown into a powerful technique for missing data.(van Buuren 2018) The bundle of grey curves indicates some possible future trajectories of the target child. The mean of the bundle is the most likely path. Graphically it is the dotted blue curve between the last measurement and the age of the outcome.

Let's look at a numerical example. We split the data into one target child and 199 donor

children, and fit a broken stick model to the donor set.

```
R> donor_data <- smocc_200 %>%
+   filter(id != "10001")
R> target_data <- smocc_200 %>%
+   filter(id == "10001" & age < 0.51)
R>
R> # fit brokenstick model at time level
R> knots <- round(c(0, 1, 2, 3, 6, 9, 12, 15, 18, 24)/12, 4)
R> fit <- brokenstick(hgt.z ~ age | id, data = donor_data,
+                     knots = knots, boundary = c(0, 3),
+                     method = "kr", seed = 15244)
```

All timepoints from the donor data enter the broken stick model. Note that the `target_data` contains only observations from the first five visits.

We now fit the prediction model on the child-level donor data. The prediction model contains the broken stick estimates for length SDS up to 6 months, as well as sex, gestational age and birth weight as covariates.

```
R> # predict with matching model at child level
R> covariates <- donor_data %>%
+   group_by(id) %>%
+   slice(1)
R> bse <- predict(fit, donor_data, x = "knots", shape = "wide")
R> donors <- bind_cols(covariates, select(bse, -id))
R> model <- lm(`1.25` ~ `0` + `0.0833` + `0.1667` + `0.25` + `0.5`
+              + sex + ga + bw, data = donors)
R> summary(model)


Call:
lm(formula = `1.25` ~ `0` + `0.0833` + `0.1667` + `0.25` + `0.5` +
    sex + ga + bw, data = donors)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3111 -0.2836  0.0034  0.2848  1.4961

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.62e-02   7.31e-01    0.02     0.98
`0`          4.18e-02   5.06e-02    0.82     0.41
`0.0833`     1.70e-02   1.00e-01    0.17     0.87
`0.1667`    -6.85e-02   1.45e-01   -0.47     0.64
`0.25`      -2.58e-01   1.41e-01   -1.82     0.07 .
`0.5`        1.17e+00   7.98e-02   14.62   <2e-16 ***
```

```
sexmale      9.16e-02   6.38e-02    1.44     0.15
ga           6.74e-03   2.22e-02    0.30     0.76
bw          -1.05e-04   9.61e-05   -1.09     0.28
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.43 on 190 degrees of freedom
Multiple R-squared:  0.774, Adjusted R-squared:  0.764
F-statistic: 81.3 on 8 and 190 DF,  p-value: <2e-16
```

The next step is to extract model predictions for both donors and target and find the ten closest donors.

```
R> donors_pred <- predict(model)
R> names(donors_pred) <- donors$id
R>
R> target <- bind_cols(
+   slice(target_data, 1),
+   select(predict(fit, target_data, x = "knots", shape = "wide"), -id))
R> target_pred <- predict(model, newdata = target)
R>
R> matches <- sort(abs(donors_pred - target_pred))[1:10]
R> matches

 11013  10093  11023  11051  10006  10068  11063  10032  11046  11080
0.0026 0.0376 0.0376 0.0693 0.0771 0.0792 0.0841 0.0852 0.0893 0.0953
```

Finally, let us study the observed and fitted trajectories of the ten matches.

```
R> ids <- as.numeric(names(matches))
R> plot(fit, new_data = donor_data, group = ids,
+      xlim = c(0, 1.4), size_y = 1, size_yhat = 0,
+      xlab = "Age (years)", ylab = "Length (SDS)",
+      ncol = 5)
```

The ten trajectories are all are close to the prediction (0.101 SD) for the target child at the age of 1.25y. Note that this does not guarantee that the histories are identical. Most matches have relatively flat curves, but a few (10051, 11023, 11086) show striking rising patterns. Nevertheless, these candidates are the best in terms of the model prediction.

If we wish the curves of the matches during the first six months to be closer to the target case, we could consider alternative metrics. A simple measure is the sum of squares differences of the broken stick estimates. Such a selection may be visually more pleasing, at the expense of prediction accuracy. On the other hand, we are less tied to setting one particular future time point, so other measures may work better when "future" is more vaguely defined as a time interval. It is still an open research question where we strike a balance. Whatever

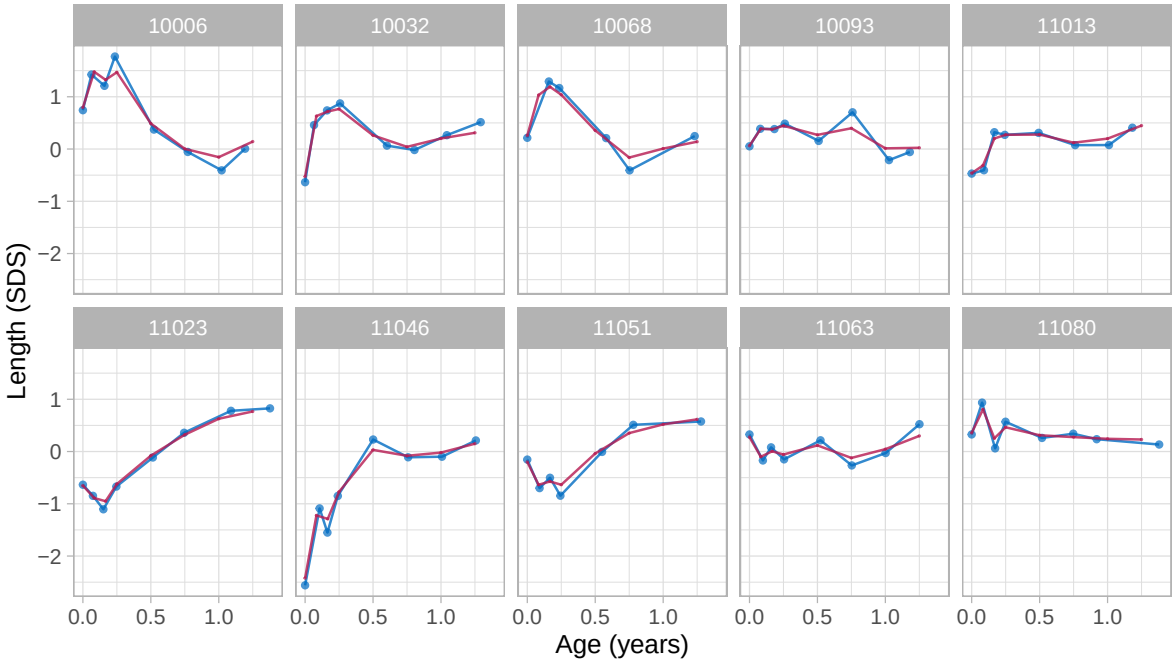

Figure 22: Curve matching. Observed and fitted trajectories of 10 matches for subject 10001.

the objectives of preferences from the user might be, the curve matching methodology, as illustrated here, has tremendous flexibility and is easy to adapt.

6. Conclusion

Overview

This paper introduces a new approach to solve the problem of irregular longitudinal data. The method absorbs the time-dependent information into a set of broken stick estimates at the subject level. The primary advantage is that it simplifies the analysis by splitting the modelling problem into two steps. First, solve the timing problem, and then solve the substantive/scientific problem. The method is mathematically simple, using a linear B-spline, conceptually simple, yet principled.

Distinctive features

The assumptions of the model cover many cases of practical interest: a straight line between breakpoints, a multivariate normal distribution for the random effects, and the MAR assumption including future data. Despite the relatively low number of model parameters, it is possible to obtain a close fit to the data, sometimes almost up to perfect reconstruction (c.f. Figure 9). There is no need to specify equidistant breakpoints. Applications in human growth and development are often more natural using non-uniformly spaced knots, which is very easy to model. Many people find it easier to understand the raw data values than the summaries. The broken stick model invites visualisation of the actual data points against time and makes it is easy to portray uncertainty as a bundle of curves. Such direct visualisation

options contribute to the explainable and responsible personalised analyses that appeal to a broad user group.

Current limitations

The broken stick model, as presented here, uses just three variables: time, measurement and group. This design choice simplifies interpretation and estimation. The lack of covariates in the model implies that the transformation from irregular data to repeated measures is identical for every subject. As long as the residual error is small, the relations with not-in-the-model variables thus remain intact. The possibility to include covariate in a second-round enhances modular modern analytic pipelines. Yet, some will prefer the direct estimation of all effects in one more extensive analysis. The current package does not support covariates. However, an experienced `R` user will have no difficulty in extending the formula in section 3.5 to include the covariates of interest.

The method that requires that all subject share the same time axis and breakpoints. In our applications, synchronisation at the start was most natural (e.g. birth, start of experiment), which is easy to do. In some cases, one might wish to anchor in the middle, e.g. at menarche, which occurs at different ages for different individuals.(Naumova, Must, and Laird 2001) It could also make sense to fasten the end, e.g. at death. However, it will be hard to do meaningful predictive analysis as we cannot anchor alive subjects. The choice of the anchor may matter less for cyclic processes. The broken stick model is not suited for applications where breakpoints vary between individuals. In those cases, it is better to use the linear mixed model directly.

Software

The Kasim-Raudenbush sampler (Kasim and Raudenbush 1998) is both fast and flexible. It produces estimates of the residual error variance per subject, can accommodate for correlation models and supports multiple imputation out-of-the-box. More research needed to establish its statistical properties especially compared to `lmer()` and other established methods. It would also be interesting to study the suitability of the correlation models implemented in the **lme4qtl** package (Ziyatdinov, Vázquez-Santiago, Brunel, Martinez-Perez, Aschard, and Soria 2018). As no training data are stored, instances of the `brokenstick` model class are tiny, often 15–20k.

Features not implemented, but that could be useful in future versions include a separate `impute()` function that inputs class `brokenstick` and returns class `mids`, a Trelliscope (Hafen and Schloerke 2020) viewer to quickly peruse hundreds of individuals model fits, an extension to multivariate time-varying and child-level data, and a generalisation to `degree > 1` to support quadratic and cubic splines.

Methodological advances

The primary modelling task for the user is to set the proper knot locations. One might envision scenarios where we want to search for the "best" places. It is not yet clear how we should do this, and how far we could automate knot placement strategies.

We need more insight into the statistical properties of procedures that execute the analysis as a sequence of steps. The relative pro's and con's of choices between multiple imputation

versus random effects are not yet fully understood.

The current procedure assumes that the within-person error is constant across all time points. However, we might expect that observing more data close to the breakpoint will reduce the uncertainty of its estimate. In some applications, we might require that the estimate should equal the observed data value when the observation time coincides with the breakpoints. While models for such scenarios are considerably more complicated, they could also increase efficiency.

Conclusion

This paper highlighted various applications of the broken stick model: critical periods, time-to-time correlation, profile analysis, curve interpolation, multiple imputation and personalised prediction. These applications certainly do not exhaust the potential of the model. My hope is that the availability of the software will stimulate creative uses, ideas and experiments.

Literature

Anderson C, Hafen R, Sofrygin O, Ryan L, HBGDki Community (2019). "Comparing predictive abilities of longitudinal child growth models." *Statistics in Medicine*, **38**(19), 3555–3570.

Argyle J, Seheult A, Wooff D (2008). "Correlation models for monitoring child growth." *Statistics in Medicine*, **27**(6), 888–904.

Bai J, Perron P (2003). "Computation and analysis of multiple structural change models." *Journal of Applied Econometrics*, **18**(1), 1–22.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.

Bellman R, Roth R (1969). "Curve fitting by segmented straight lines." *Journal of the American Statistical Association*, **64**(327), 1079–1084.

Cole T (1995). "Conditional reference charts to assess weight gain in British infants." *Archives of Disease in Childhood*, **73**(1), 8–16.

de Boor C (1978). *A practical guide to splines*. Springer-Verlag, New York.

de Kroon M, Renders C, van Wouwe J, van Buuren S, Hirasing R (2010). "The Terneuzen birth cohort: BMI changes between 2 and 6 years correlate strongest with adult overweight." *PloS ONE*, **5**(2), e9155.

de Kroon MLA, Renders CM, Kuipers EC, van Wouwe JP, van Buuren S, de Jonge GA, Hirasing RA (2008). "Identifying metabolic syndrome without blood tests in young adults - The Terneuzen birth cohort." *European Journal of Public Health*, **18**(6), 656–660.

Dietz W (1994). "Critical periods in childhood for the development of obesity." *American Journal of Clinical Nutrition*, **59**(5), 955–959.

Diggle P (1988). "An approach to the analysis of repeated measurements." *Biometrics*, pp. 959–971.

Diggle P, Heagerty P, Liang K, Zeger S (2002). *Analysis of longitudinal data. Second Edition.* Oxford University Press, Oxford.

Fitzmaurice GM, Laird NM, Ware JH (2011). *Applied longitudinal analysis. Second edition.* John Wiley & Sons, New York.

Fredriks A, van Buuren S, Burgmeijer R, Meulmeester J, Beuker R, Brugman E, Roede M, Verloove-Vanhorick S, Wit J (2000). "Continuing positive secular growth change in The Netherlands 1955-1997." *Pediatric Research*, **47**(3), 316–323.

Gelman A, Hill J (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge University Press, Cambridge.

Hafen R, Schloerke B (2020). *trelliscopejs: Create Interactive Trelliscope Displays.* R package version 0.2.5, URL https://CRAN.R-project.org/package=trelliscopejs.

Hand D, Taylor C (1987). *Multivariate analysis of variance and repeated measures: A practical approach for behavioural scientists.* CRC press, Boca Raton, FL.

Herngreen WP, van Buuren S, van Wieringen JC, Reerink JD, Verloove-Vanhorick SP, Ruys JH (1994). "Growth in length and weight from birth to 2 years of a representative sample of Netherlands children (born in 1988-89) related to socio-economic status and other background characteristics." *Annals of Human Biology*, **21**(5), 449–463.

Ismail L, Puglia F, Ohuma E, Ash S, Bishop D, Carew R, Al Dhaheri A, Chumlea W (2016). "Precision of recumbent crown-heel length when using an infantometer." *BMC Pediatrics*, **16**(1), 186.

Johnson R, Wichern D (1988). *Applied multivariate statistical analysis. Second Edition.* Prentice Hall, Englewood Cliffs, NJ.

Kasim RM, Raudenbush SW (1998). "Application of Gibbs sampling to nested variance components models with heterogeneous within-group variance." *Journal of Educational and Behavioral Statistics*, **23**(2), 93–116.

Kenward M (1987). "A method for comparing profiles of repeated measurements." *Journal of the Royal Statistical Society C (Applied Statistics)*, **36**(3), 296–308.

Koutsoyiannis D (2000). "Broken line smoothing: a simple method for interpolating and smoothing data series." *Environmental Modelling & Software*, **15**(2), 139–149.

Krone T, Boessen R, Bijlsma S, van Stokkum R, Clabbers N, Pasman W (2020). "The possibilities of the use of N-of-1 and do-it-yourself trials in nutritional research." *PloS ONE*, **15**(5), e0232680.

Laird NM, Ware JH (1982). "Random-effects models for longitudinal data." *Biometrics*, **38**(4), 963–974.

Lerman P (1980). "Fitting segmented regression models by grid search." *Journal of the Royal Statistical Society C (Applied Statistics)*, **29**(1), 77–84.

Lokku A, Lim L, Birken C, Pullenayegum E, TARGet Kids! Collaboration, (2020). "Summarizing the extent of visit irregularity in longitudinal data." *BMC Medical Research Methodology*, **20**, 1–9.

MacArthur R (1957). "On the relative abundance of bird species." *Proceedings of the National Academy of Sciences of the United States of America*, **43**(3), 293.

Morrison D (1976). *Multivariate statistical methods. Second edition.* McGraw-Hill, Singapore.

Naumova EN, Must A, Laird NM (2001). "Tutorial in Biostatistics: Evaluating the impact of 'critical periods' in longitudinal studies of growth using piecewise mixed effects models." *International Journal of Epidemiology*, **30**, 1332–1341.

Pullenayegum E, Lim L (2016). "Longitudinal data subject to irregular observation: A review of methods with a focus on visit processes, assumptions, and study design." *Statistical Methods in Medical Research*, **25**(6), 2992–3014.

R Core Team (2020). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rehfeld K, Marwan N, Heitzig J, Kurths J (2011). "Comparison of correlation analysis techniques for irregularly sampled time series." *Nonlinear Processes in Geophysics*, **18**(3), 389–404.

Rubin DB (1987). *Multiple Imputation for Nonresponse in Surveys.* John Wiley & Sons, New York.

Skrondal A, Rabe-Hesketh S (2009). "Prediction in multilevel generalized linear models." *Journal of the Royal Statistical Society A (Statistics in Society)*, **172**(3), 659–687.

Toms J, Lesperance M (2003). "Piecewise regression: A tool for identifying ecological thresholds." *Ecology*, **84**(8), 2034–2041.

Towers S (2014). "Potential fitting biases resulting from grouping data into variable width bins." *Physics Letters B*, **735**, 146–148.

van Buuren S (2007). "Growth references." In C Kelnar, M Savage, P Saenger, C Cowell (eds.), *Growth Disorders 2nd*, pp. 165–181. Hodder Arnold, London.

van Buuren S (2014). "Curve matching: A data-driven technique to improve individual prediction of childhood growth." *Annals of Nutrition & Metabolism*, **65**(3), 227–233.

van Buuren S (2018). *Flexible Imputation of Missing Data. Second Edition.* CRC Press, Boca Raton, FL.

Vaughan D, Kuhn M (2020). *hardhat: Construct Modeling Packages.* R package version 0.1.4, URL https://CRAN.R-project.org/package=hardhat.

Ziyatdinov A, Vázquez-Santiago M, Brunel H, Martinez-Perez A, Aschard H, Soria J (2018). "lme4qtl: Linear mixed models with flexible covariance structure for genetic studies of related individuals." *BMC Bioinformatics*, **19**(1), 1–5.

Affiliation:

Stef van Buuren
Netherlands Organisation for Applied Scientific Research TNO & University of Utrecht
Schipholweg 77 2316 ZL Leiden
E-mail: stef.vanbuuren@tno.nl
URL: https://stefvanbuuren.name

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd