

Online Appendix to accompany: Sterba, S.K. (In Press). Fitting nonlinear latent growth curve models with individually-varying time points. *Structural Equation Modeling*.

Contents:

I. *Mplus* Input for shape-factor LGM with balanced time points (original and re-expression).....p.2

II. *Mplus* Input for shape-factor LGM with individually-varying time points.....p.3

III. %PIECEWISE SAS Macro for transforming time scores to have desired piecewise structure, for piecewise LGMs.....p.4

IV. User instructions, input, and examples for %PIECEWISE Macro with balanced (Case I or II) time points.....p.6

V. User instructions, input, and examples for %PIECEWISE Macro with individually-varying (Case III) time points.....p.11

VI. *Mplus* Input for empirical example fitting linear, quadratic, shape-factor, piecewise-linear, piecewise-quadratic, and exponential LGMs with individually-varying time points.....p. 16

I. Mplus input for shape-factor LGM with balanced time points

Original expression

```
DATA: file is yourdata.dat;
VARIABLE: Names are y1-y7;
ANALYSIS: estimator=ml;
MODEL:
int by y1-y7@1;
shape by y1@0
y2-y6*
y7@6;
y1-y7 (1);
[y1-y7@0];
[int shape];
int shape;
int with shape;
```

Re-expression

```
DATA: file is yourdata.dat;
VARIABLE: Names are y1-y7;
ANALYSIS: estimator=ml;
MODEL:
int by y1-y7@1;
shape by
y1@0
y2*(L1)
y3*(L2)
y4*(L3)
y5*(L4)
y6*(L5)
y7@6;
y1-y7 (1);
[y1-y7@0];
[int shape];
int shape;
int with shape;
MODEL CONSTRAINT:
NEW(d1-d5);
L1=1+d1;
L2=2+d2;
L3=3+d3;
L4=4+d4;
L5=5+d5;
```

II. *Mplus* Input for shape-factor LGM with individually-varying time points

```
DATA: file is yourdata.dat;
VARIABLE:
Names are y1-y7 time0-time6;
Usevariables are y1-y7;
constraint=time0-time6;
ANALYSIS: estimator=ml;
MODEL:
int by y1-y7@1;
shape by y1*(t0)
y2 (t1)
y3 (t2)
y4 (t3)
y5 (t4)
y6 (t5)
y7 (t6);
y1-y7 (1);
[y1-y7@0];
[int shape];
int shape;
int with shape;
MODEL CONSTRAINT:
NEW(d1-d5);
t0=time0;
t1=time1+d1;
t2=time2+d2;
t3=time3+d3;
t4=time4+d4;
t5=time5+d5;
t6=time6;
```

III. %PIECEWISE SAS Macro for transforming time scores to have desired piecewise structure, for piecewise LGMs

Below is the SAS Macro %PIECEWISE, which implements Table 1's general guidelines for transforming (already-recoded) time scores to have desired piecewise structure. Users should read this code into SAS.

```
%macro piecewise;
%let knot0=. ;
%let last=%eval(&knots+1);
%let knot&last=. ;
%do m=1 %to &phases. ;
%let n=%eval(&m-1);
data yourdata; set yourdata;
if &m < &interceptphase. then do;
    if (&&knot&m. >= time) and ((time > &&knot&n.) or
        (&&knot&n. =.)) then loadphase&m.=(time-&&knot&m.);
else if (&&knot&n. ^=.) and (&&knot&n. >= time) then loadphase&m. =(&&knot&n.-&&knot&m.);
else loadphase&m.=0;
if time=. then loadphase&m.=.; /*optional line -- missing (.) time score given . loading*/
end;

if &m=&interceptphase. then do;
if ((&&knot&m. >= time) or (&&knot&m. =.)) and
((time > &&knot&n.) or (&&knot&n. =.)) then loadphase&m.=time;
else if &&knot&n. >= time then loadphase&m.=&&knot&n.;
else if &&knot&m. < time then loadphase&m. =&&knot&m.;
if time=. then loadphase&m.=.; /*optional line -- missing (.) time score given . loading*/
end;

if &m > &interceptphase. then do;
if (&&knot&n. < time) and ((time <= &&knot&m.) or (&&knot&m.=.)) then loadphase&m.=(time-
&&knot&n.);
else if (&&knot&m. ^=.) and (&&knot&m. < time) then loadphase&m.=(&&knot&m.-&&knot&n.);
else loadphase&m.=0;
if time=. then loadphase&m.=.; /*optional line -- missing (.) time score given . loading*/
end;
run;
%end;
proc print data=yourdata; run;
%mend;
```

After reading in the macro, users will need to supply some input before executing the macro. In this input users specify (i) their (already-recoded) time scores, (ii) their desired number of phases (M), (iii) their desired number of knots ($M-1$), (iv) the *intercept phase*, and (v) the (already-recoded) time score corresponding with each knot. (Recall that the *intercept phase* is the phase containing the origin of time (in the already-recoded time metric)). The Macro will then generate linear slope loadings for a piecewise LGM and print these to the output window. If a researcher has higher-order polynomial pieces (quadratic, cubic, etc.), they have the option of specifying that columns of lambda be constrained to powered linear loadings during model specification (see *Mplus* empirical example code for piecewise quadratic, discussed later). Specifying the input is illustrated by example. In particular, in the next section we provide input for three balanced case examples (one of which is described in the manuscript also). Users will need to tailor the input code to their particular analysis.

Online Appendix

Note that there are three optional lines of Macro code (denoted as such with green comments) that set piecewise loadings to be missing (.) when original time scores are missing (.). If the researcher is using balanced (Case I or II) time scores, which are assumed nonmissing for everyone, this is irrelevant. If the researcher is using individually varying time scores (Case III), but has no missing time scores, this is irrelevant. If the researcher is using individually varying time scores (Case III), and has missing data, this is relevant and will be discussed in section V.

IV. User instructions, input, and examples for %PIECEWISE Macro with balanced (Case I or II) time points

Input code for Balanced Case Piecewise Example 1. In Example 1, the already-recorded time scores in a sample are $time=0, 1, 2, 3, 4, 5, 6$. There are $M=2$ phases (pieces) with $M-1=1$ knot which the researcher wants to place at $time=3$. The *intercept phase* would be phase 1 because $time=0$ falls before the knot of $time=3$. To generate transformed time scores for Example 1, specify:

```

DATA yourdata;
INPUT time;
CARDS;
0
1
2
3
4
5
6
;
RUN;
%let phases=2;
%let knots=1;
%let interceptphase=1;
%let knot1=3;
%piecewise;
    
```

The macro will expect the already-recorded time scores to be in a variable called **time** in a temporary SAS dataset called **yourdata**. [The above code shows one way of making such a dataset: in a **DATA** statement, a temporary SAS dataset called **yourdata** is initialized, and it contains time scores. This temporary dataset contains one **INPUT** variable (**time**). The user will need to list their T values of **time**, one per row, after the **CARDS** command.] If the user has their time scores in a permanent dataset, or a temporary dataset of another name, they will have to rename their dataset. If the user has their time scores in a variable labeled something other than **time**, it will also need to be renamed.

%let phases= Here the user specifies the desired number of phases (pieces, i.e., M). Here, 2.
%let knots= Here the user specifies the desired number of knots ($M-1$). Here, 1.
%let interceptphase= Here the user identifies what phase contains the intercept of time. Here, 1.
%let knot1= Since the user here had one knot and desired to place it at $time=3$, specify $knot1=3$.
%piecewise; Invokes (executes) the Macro. This line should only be run after the other preliminary input code has been run.

Output window will show:

loadphase1	loadphase2
0	0
1	0
2	0
3	0
3	1
3	2
3	3

The loadphase1 and loadphase2 columns from the output window can be directly inserted as linear slope loadings for a 2-piece linear spline LGM:

$$\Lambda = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 0 \\ 1 & 3 & 0 \\ 1 & 3 & 1 \\ 1 & 3 & 2 \\ 1 & 3 & 3 \end{pmatrix}$$

Note that these piecewise linear loadings sum across rows to equal the linear time scores 0, 1, 2, 3, 4, 5, 6. If one or more quadratic or cubic pieces are desired these linear slope loadings can be squared or cubed accordingly. This is done using a nonlinear constraint in the empirical example code for a piecewise quadratic LGM, further below.

Input code for Balanced Case Piecewise Example 2 (described in manuscript). In Example 2, the already-recoded time scores in a sample are $time = -6.5 -5, -1, 0, 2, 3.5, 6, 8$. (Note that the time scores are now unequally spaced, though still balanced in the sense that--should persons have no missing data--they would all be measured at the same times.) There are $M=3$ phases (pieces), and $M-1=2$ knots occurring at $time = -1$, and 3.5 . The *intercept phase* is at phase 2 because $time=0$ falls in between the two knots. To generate transformed time scores for Example 2, specify:

```
DATA yourdata;
INPUT TIME;
CARDS;
-6.5
-5
-1
0
2
3.5
6
8
;
RUN;
%let phases=3;
%let knots=2;
%let interceptphase=2;
%let knot1=-1; %let knot2=3.5;
%piecewise;
```

`%let phases=` Here the researcher desires the number of phases (M) to be 3.

`%let knots=` Here the desired number of knots ($M-1$) is 2.

`%let interceptphase=` Here 2, since 0 falls between knot1 and knot2.

`%let knot1=` Since the user desired to place the first knot at $time = -1$, specify `knot1 = -1`.

`%let knot2=` Since the user desired to place the second knot at $time = 3.5$, specify `knot2 = 3.5`.

Output window will show:

loadphase1	loadphase2	loadphase3
-5.5	-1.0	0.0
-4.0	-1.0	0.0
0.0	-1.0	0.0
0.0	0.0	0.0
0.0	2.0	0.0
0.0	3.5	0.0
0.0	3.5	2.5
0.0	3.5	4.5

The loadphase1, loadphase2, and loadphase3 columns from the output window can be directly inserted as linear slope loadings for a 3-piece linear spline LGM:

$$\Lambda = \begin{pmatrix} 1 & -5.5 & -1 & 0 \\ 1 & -4 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 0 & 3.5 & 0 \\ 1 & 0 & 3.5 & 2.5 \\ 1 & 0 & 3.5 & 4.5 \end{pmatrix}$$

Note that these piecewise linear loadings sum across rows to equal the linear time scores -6.5 -5, -1, 0, 2, 3.5, 6, 8. If one or more quadratic or cubic pieces are desired these linear slope loadings can be squared or cubed accordingly.

Input code for Balanced Case Piecewise Example 3. In Example 3, suppose that a study's already-recorded time scores are *time*=-30, -27, -24, -23, -22, -19, -14, -3, 0, 4, 7, 11, 18, 22, 24, 27, 30, 33, (again, unequally spaced). *M*=6 phases (pieces). *M*-1=5 knots occurring at *time* = -24, -19, 0, 18, 27. Hence, the *intercept phase* would be phase 3 because *time*=0 falls at a boundary between the third and fourth phases, and is thus assigned to the previous phase. To generate transformed time scores for Example 3, specify:

```
DATA yourdata;
INPUT time;
CARDS;
-30
-27
-24
-23
-22
-19
-14
-3
0
```

Online Appendix

```

4
7
11
18
22
24
27
30
33
;
RUN;
%let phases=6;
%let knots=5;
%let interceptphase=3;
%let knot1=-24; %let knot2=-19; %let knot3=0; %let knot4=18;
%let knot5=27;
%piecewise;

```

`%let phases=` Here the researcher desires the number of phases (M) to be 6.

`%let knots=` Here the desired number of knots ($M-1$) is 5.

`%let interceptphase=` Here 3, since the Table 1 guidelines state that when an intercept occurs at a knot, it is assigned to the previous phase.

`%let knot1=` Since the user desired to place the first knot at $time=-24$, specify `knot1=-24`.

`%let knot2=` Since the user desired to place the second knot at $time=-19$, specify `knot2=-19`.

`%let knot3=` Since the user desired to place the third knot at $time=0$, specify `knot3=0`.

`%let knot4=` Since the user desired to place the fourth knot at $time=18$, specify `knot4=18`.

`%let knot5=` Since the user desired to place the fifth knot at $time=27$, specify `knot5=27`.

Output window will show:

loadphase1	loadphase2	loadphase3	loadphase4	loadphase5	loadphase6
-6	-5	-19	0	0	0
-3	-5	-19	0	0	0
0	-5	-19	0	0	0
0	-4	-19	0	0	0
0	-3	-19	0	0	0
0	0	-19	0	0	0
0	0	-14	0	0	0
0	0	-3	0	0	0
0	0	0	0	0	0
0	0	0	4	0	0
0	0	0	7	0	0
0	0	0	11	0	0
0	0	0	18	0	0
0	0	0	18	4	0
0	0	0	18	6	0
0	0	0	18	9	0
0	0	0	18	9	3
0	0	0	18	9	6

The loadphase1, loadphase2, and loadphase3, loadphase4, loadphase5, and loadphase6 columns from the output window can be directly inserted as linear slope loadings for a 3-piece linear spline LGM :

Online Appendix

$$\mathbf{\Lambda} = \begin{pmatrix} 1 & -6 & -5 & -19 & 0 & 0 & 0 \\ 1 & -3 & -5 & -19 & 0 & 0 & 0 \\ 1 & 0 & -5 & -19 & 0 & 0 & 0 \\ 1 & 0 & -4 & -19 & 0 & 0 & 0 \\ 1 & 0 & -3 & -19 & 0 & 0 & 0 \\ 1 & 0 & 0 & -19 & 0 & 0 & 0 \\ 1 & 0 & 0 & -14 & 0 & 0 & 0 \\ 1 & 0 & 0 & -3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 4 & 0 & 0 \\ 1 & 0 & 0 & 0 & 7 & 0 & 0 \\ 1 & 0 & 0 & 0 & 11 & 0 & 0 \\ 1 & 0 & 0 & 0 & 18 & 0 & 0 \\ 1 & 0 & 0 & 0 & 18 & 4 & 0 \\ 1 & 0 & 0 & 0 & 18 & 6 & 0 \\ 1 & 0 & 0 & 0 & 18 & 9 & 0 \\ 1 & 0 & 0 & 0 & 18 & 9 & 3 \\ 1 & 0 & 0 & 0 & 18 & 9 & 6 \end{pmatrix}$$

Note again that the linear slope loadings sum across rows to equal the time scores. If one or more quadratic or cubic pieces are desired these linear slope loadings can be squared or cubed accordingly.

V. User instructions, input, and examples for %PIECEWISE Macro with individually-varying time points

We can use the same %PIECEWISE Macro described earlier (section III) to generate Λ_i for the individually-varying time scores case. However, in this case instead of inputting one set of (already-recoded) time scores that apply to all persons in the sample, we input an entire dataset of N persons' (already-recoded) time scores.

Requirements: The four important requirements for the input dataset are as follows.

(1) First, this input dataset of time scores must be in *long format*. See Willett and Singer (2003) for details on long vs. wide dataset formats. To illustrate long format, here is an excerpt of the input empirical example dataset from the manuscript. Note that a given case (*id*) has multiple records, one for each repeated measurement. Note that the dataset submitted to %PIECEWISE does not need to contain only the time scores per occasion (i.e. the column labeled *time*). The long format dataset can and should also contain other variables that the researcher intends to use in their LGM analysis--here, for instance *outcome* (our y -scores).

	id	occasion	time	outcome
18397	1500090	0	0	2.49476
18398	1500090	1	0.1644736842	4
18399	1500090	2	0.3371710526	4.8
18400	1500090	3	0.586622807	5.9
18401	1500090	4	0.6661184211	6.1
18402	1500090	5	0.8278508772	6.6
18403	1500090	6	0.9950657895	7.1
18404	300154	0	0	2.85
18405	300154	1	0.1644736842	5
18406	300154	2	0.3344298246	6.3
18407	300154	3	0.5921052632	7.2
18408	300154	4	0.7346491228	8
18409	300154	5	0.8360745614	7.4
18410	300154	6	.	.
18411	600121	0	0	3.8
18412	600121	1	0.1672149123	5.6
18413	600121	2	0.3207236842	7.65
18414	600121	3	0.5948464912	9.7
18415	600121	4	0.6716008772	9.7
18416	600121	5	0.8251096491	9.8
18417	600121	6	1.0032894737	9.7
18418	2800173	0	0	2.9
18419	2800173	1	0.1644736842	4.55
18420	2800173	2	0.3262061404	6
18421	2800173	3	0.6003289474	6.5

(2) Second, %PIECEWISE will expect the long format input dataset to be a temporary SAS dataset called **yourdata**; if the user has their time scores in a permanent dataset, or in a temporary dataset of another name, the dataset will have to be renamed in this manner. For instance, here is an example of

reading a text dataset saved on the C-drive root folder, and called original.dat, into a SAS temporary dataset called **yourdata**. The dataset contains the variables displayed in the above excerpt.

```
data yourdata;
infile "C:\original.dat";
input id occasion time outcome; run;
```

(3) Third, %PIECEWISE will also expect the individually-varying time scores to be in a variable called **time** (as above). Any other variables included in the dataset can have arbitrary names.

(4) Fourth, missing time scores require special consideration under Case III. Note that there are three optional lines of Macro code (denoted as such with green comments) that set piecewise loadings to be missing (.) when original time scores are missing (.). If the researcher is using individually varying time scores (Case III) and has missing data, this is relevant and will be discussed here.

- *Missing data codes for the Macro.* If the researcher's original dataset has missing time scores denoted with a symbol other than a period (.) then either missing values should be recoded to a period (.) before using the Macro, or the macro code should be modified to use the existing missing code instead of the period (.).
- *Missing data codes for Mplus.* If a period (.) is used as the missing data code for piecewise loadings in the analysis dataset, and (.) is given as the missing data code in Mplus (MISSING=.;), the default will be for Mplus to delete cases having at least one missing piecewise loading value. As described in the manuscript, this default is in place when fitting a LGM with a conditional likelihood (i.e., likelihood conditional on predictors), since in Case III time scores are considered observed predictors. There are two different ways to override this default. One approach is appropriate *when time scores and y scores are both missing at the same occasion(s)*. When observations are missing both time scores and y-scores, the researcher can insert any arbitrary numeric value as a placeholder for the time score at occasions where y-scores are missing (so long as the placeholder is not what is given to Mplus as a missing data code on the MISSING= line). This will serve to retain the entire person (dataset row). But only that person's *observed* (nonmissing) y-scores--along with their paired, true/nonarbitrary time scores--will contribute to model estimation and resulting parameter estimates. For instance, a researcher who is using (.) as their missing data code in Mplus could insert 999 for time scores associated with missing y scores. Code for doing so is demonstrated further below for a piecewise linear LGM with 2 pieces. The other approach is appropriate *when time score(s) are missing for occasion(s) at which y-score(s) are present*. This situation is expected to be less common in practice. This situation cannot be addressed by arbitrary placeholders of, say, 999 as described above. Rather, time scores would ideally need to be multiply imputed (less ideal is to impute with mean times for each occasion window).

Online Appendix

Input: After supplying information on the M phases, $M-1$ knots, *time* at knot point(s), and *intercept phase* (see balanced case section for definitions), executing %PIECEWISE will add M new columns of transformed linear slope loading values to the dataset.

For instance, for the empirical example in the manuscript we desired $M=2$ phases (pieces), and $M-1=1$ knots occurring at *time* = .5 (which corresponded to 6 months of age in our infant weight example). Recall that we are assuming that the knot occurs at the same time for all persons. Our *intercept phase* would be phase 1 because *time*=0 falls in the first phase. To generate transformed time scores for this empirical example with individually-varying measurements, specify:

```
%let phases=2;
%let knots=1;
%let interceptphase=1;
%let knot1=.5;
%piecewise;
```

`%let phases=` Where the researcher specifies the desired number of phases (M), here 2.
`%let knots=` Where the researcher specifies the desired number of knots ($M-1$), here 1.
`%let interceptphase=` Here 1, since 0 occurs in the first phase.

Output: After running %PIECEWISE, the dataset **yourdata** will contain M extra columns corresponding to the linear slope loadings for your M pieces. Here is an excerpt of what **yourdata** looks like for the empirical example after running %PIECEWISE.

	id	occasion	time	outcome	loadphase1	loadphase2
18397	1500090	0	0	2.49476	0	0
18398	1500090	1	0.1644736842	4	0.16447	0
18399	1500090	2	0.3371710526	4.8	0.33717	0
18400	1500090	3	0.586622807	5.9	0.5	0.08662
18401	1500090	4	0.6661184211	6.1	0.5	0.16612
18402	1500090	5	0.8278508772	6.6	0.5	0.32785
18403	1500090	6	0.9950657895	7.1	0.5	0.49507
18404	300154	0	0	2.85	0	0
18405	300154	1	0.1644736842	5	0.16447	0
18406	300154	2	0.3344298246	6.3	0.33443	0
18407	300154	3	0.5921052632	7.2	0.5	0.09211
18408	300154	4	0.7346491228	8	0.5	0.23465
18409	300154	5	0.8360745614	7.4	0.5	0.33607
18410	300154	6
18411	600121	0	0	3.8	0	0
18412	600121	1	0.1672149123	5.6	0.16721	0
18413	600121	2	0.3207236842	7.65	0.32072	0
18414	600121	3	0.5948464912	9.7	0.5	0.09485
18415	600121	4	0.6716008772	9.7	0.5	0.1716
18416	600121	5	0.8251096491	9.8	0.5	0.32511
18417	600121	6	1.0032894737	9.7	0.5	0.50329
18418	2800173	0	0	2.9	0	0
18419	2800173	1	0.1644736842	4.55	0.16447	0
18420	2800173	2	0.3262061404	6	0.32621	0
18421	2800173	3	0.6003289474	6.5	0.5	0.10033

Online Appendix

The M extra columns contain piecewise linear LGM slope loadings for N Λ_i , stacked on top of each other. For instance, this excerpt implies that, for a two-piece piecewise linear LGM, Λ_i for ID# 1500090 would look like:

$$\Lambda_{1500090} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & .164 & 0 \\ 1 & .337 & 0 \\ 1 & .5 & .087 \\ 1 & .5 & .166 \\ 1 & .5 & .328 \\ 1 & .5 & .495 \end{pmatrix}$$

And Λ_i for ID# 300154 would look like:

$$\Lambda_{300154} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & .164 & 0 \\ 1 & .334 & 0 \\ 1 & .5 & .092 \\ 1 & .5 & .235 \\ 1 & .5 & .336 \\ 1 & . & . \end{pmatrix}$$

As can be seen in the above dataset, ID#300154 was missing their final outcome measurement as well as their final time score. The Macro thus assigned this person to have missing piecewise loadings at their last measurement occasion in Λ_{300154} . As discussed earlier in the missing data section, when both y -scores and time scores for an observation are missing, the researcher will want to insert an arbitrary placeholder for the time score (that is something other than the general code used to denote missingness in the analysis dataset). This will prevent Mplus from listwise deleting the entire case. This is done for the empirical example piecewise linear LGM model with 2 pieces, using the below code; missing piecewise loadings are given an arbitrary placeholder of 999.

```
data yourdata; set yourdata;
if loadphase1=. then loadphase1=999;
if loadphase2=. then loadphase2=999; run;
```

The researcher should consult the manuscript for instruction on other kinds of missingness (i.e., missingness on an observation's y -score, but not its associated time score; missingness on an observation's time score, but not its associated y -score).

Online Appendix

Although in *long format* it is easy to visualize what the $N \Lambda_i$ would look like, as they are all stacked on top of each other, SEM software used to estimate LGMs will expect the transformed, individually-varying LGM time scores serving as definition variables to be in *wide format*.¹ It is most convenient to output this dataset in long format, and transfer it to wide format in *Mplus* immediately prior to model fitting. Example *Mplus* piecewise LGM code is given below that transforms long-to-wide in the same syntax file as used for model fitting. (It is also possible to transform wide to long within SAS, e.g. using Proc Transpose, but this is not discussed here). Hence, as a last step we will simply output our long format dataset **yourdata** (which contains all analysis variables, including transformed time scores) as a text file so that it can be read by *Mplus*. For the empirical data example, this last step looks like:

```
data _NULL_; set yourdata;
file "C:\yourpiecisedata.dat";
put id occasion outcome loadphase1 loadphase2; run;
```

¹ In wide format, our unconditional piecewise LGM dataset will have $T+TM$ columns (T repeated measures + TM transformed linear time scores), even if some pieces have higher-order slope factors. For pieces with higher order (e.g. quadratic or cubic) slopes, their respective factor loadings can be specified as squares or cubes of the linear time scores in the dataset, using nonlinear constraints in the model specification (see empirical example code).

VI. Mplus Input for empirical example fitting linear, quadratic, shape-factor, piecewise-linear, piecewise-quadratic, and exponential LGMs with individually-varying time points

Note: See manuscript for a discussion of handling missingness on time scores. Section V. above also has some specific information on missing data handling that relates to the use of the %PIECWISE macro for piecewise LGMs.

Linear LGM with individually-varying time scores: Empirical example code

```

DATA: file is yourdata.dat;
VARIABLE: Names are id y1-y7 time0-time6;
Usevariables are y1-y7;
constraint=time0-time6;
missing=.;
ANALYSIS: estimator=ml;
MODEL:
int by y1-y7@1 ;
linear by y1*(t0)
y2 (t1)
y3 (t2)
y4 (t3)
y5 (t4)
y6 (t5)
y7 (t6);
y1-y7;
[y1-y7@0];
[int linear];
int linear;
int with linear;

MODEL CONSTRAINT:
t0=time0;
t1=time1;
t2=time2;
t3=time3;
t4=time4;
t5=time5;
t6=time6;

```

Note that Mplus can also use a TSCORES option (rather than a MODEL CONSTRAINT option) to fit polynomial growth models with individually-varying time-points, but this alternative is not illustrated here.

Quadratic LGM with individually-varying time scores: Empirical example code

```

DATA: file is yourdata.dat;
VARIABLE: Names are id y1-y7 time0-time6;
Usevariables are y1-y7;
  constraint=time0-time6;
  missing=.;
ANALYSIS: estimator=ml;
MODEL:
int by y1-y7@1;
linear by y1*(L0)
y2 (L1)
y3 (L2)
y4 (L3)
y5 (L4)
y6 (L5)
y7 (L6);
quad by y1*(Q0)
y2 (Q1)
y3 (Q2)
y4 (Q3)
y5 (Q4)
y6 (Q5)
y7 (Q6);
y1-y7;
[y1-y7@0];
[int linear quad];
int linear quad;
int linear quad with int linear quad;

MODEL CONSTRAINT:
L0=time0;
L1=time1;
L2=time2;
L3=time3;
L4=time4;
L5=time5;
L6=time6;
Q0=time0*time0;
Q1=time1*time1;
Q2=time2*time2;
Q3=time3*time3;
Q4=time4*time4;
Q5=time5*time5;
Q6=time6*time6;

```

Note that Mplus can also use a TSCORES option (rather than a MODEL CONSTRAINT option) to fit polynomial growth models with individually-varying time-points, but this alternative is not illustrated here.

Shape-factor LGM (re-expressed) with indiv.-varying time scores: Empirical example code

```
DATA: file is yourdata.dat;
VARIABLE: Names are id y1-y7 time0-time6;
Usevariables are y1-y7;
  constraint=time0-time6;
  missing=.;
ANALYSIS: estimator=ml;
MODEL:
  int by y1-y7@1;
  shape by y1*(t0)
  y2 (t1)
  y3 (t2)
  y4 (t3)
  y5 (t4)
  y6 (t5)
  y7 (t6);
  y1-y7;
  [y1-y7@0];
  [int shape];
  int shape;
  int with shape;

MODEL CONSTRAINT:
  NEW(d1-d5);
  t0=time0;
  t1=time1+d1;
  t2=time2+d2;
  t3=time3+d3;
  t4=time4+d4;
  t5=time5+d5;
  t6=time6;
```

Piecewise Linear LGM with individually-varying time scores: Empirical example code with long format input

****Including code to restructure data from long format to wide format because
%PIECEWISE Macro output is in long format but Mplus LGM code expects wide format****

```

DATA: file is yourpiecewisedata.dat;
DATA LONGTOWIDE:
LONG = outcome | loadphase1 | loadphase2; !long format (original)
WIDE = y1 y2 y3 y4 y5 y6 y7 |
phase1time0 phase1time1 phase1time2 phase1time3
phase1time4 phase1time5 phase1time6 |
phase2time0 phase2time1 phase2time2
phase2time3 phase2time4 phase2time5 phase2time6; !wide format
IDVARIABLE = id;
REPETITION =occasion(0 1 2 3 4 5 6);

VARIABLE: Names are id occasion outcome
loadphase1 loadphase2; !long format (original)

Usevariables are y1-y7; !wide format

constraint= phase1time0 phase1time1
phase1time2 phase1time3 phase1time4
phase1time5 phase1time6 phase2time0
phase2time1 phase2time2 phase2time3
phase2time4 phase2time5 phase2time6; !wide format

missing=.;
ANALYSIS: estimator=ml;
MODEL:
int by y1-y7@1;
linearp1 by y1*(LP10)
y2 (LP11)
y3 (LP12)
y4 (LP13)
y5 (LP14)
y6 (LP15)
y7 (LP16);
linearp2 by y1*(LP20)
y2 (LP21)
y3 (LP22)
y4 (LP23)
y5 (LP24)
y6 (LP25)
y7 (LP26);
y1-y7;
[y1-y7@0];
[int linearp1 linearp2];
int linearp1 linearp2;
int linearp1 linearp2 with int linearp1 linearp2;

MODEL CONSTRAINT:
LP10=phase1time0;

```

Online Appendix

LP11=phase1time1;
LP12=phase1time2;
LP13=phase1time3;
LP14=phase1time4;
LP15=phase1time5;
LP16=phase1time6;

LP20=phase2time0;
LP21=phase2time1;
LP22=phase2time2;
LP23=phase2time3;
LP24=phase2time4;
LP25=phase2time5;
LP26=phase2time6;

Piecewise Linear LGM with individually-varying time scores: Empirical example code with wide format input

****This code assumes you have already taken the long format %PIECEWISE Macro output and restructured it to wide format outside of *Mplus*****

```

DATA: file is yourwidepiecedata.dat;
VARIABLE:
Names are id phase1time0-phase1time6
phase2time0-phase2time6 y1-y7;
Usevariables are y1-y7;
constraint= phase1time0-phase1time6
phase2time0-phase2time6;
missing=.;
ANALYSIS: estimator=ml;

MODEL:
int by y1-y7@1;
linearp1 by y1*(LP10)
y2 (LP11)
y3 (LP12)
y4 (LP13)
y5 (LP14)
y6 (LP15)
y7 (LP16);
linearp2 by y1*(LP20)
y2 (LP21)
y3 (LP22)
y4 (LP23)
y5 (LP24)
y6 (LP25)
y7 (LP26);
y1-y7;
[y1-y7@0];
[int linearp1 linearp2];
int linearp1 linearp2;
int linearp1 linearp2 with int linearp1 linearp2;

MODEL CONSTRAINT:
LP10=phase1time0;
LP11=phase1time1;
LP12=phase1time2;
LP13=phase1time3;
LP14=phase1time4;
LP15=phase1time5;
LP16=phase1time6;
LP20=phase2time0;
LP21=phase2time1;
LP22=phase2time2;
LP23=phase2time3;
LP24=phase2time4;
LP25=phase2time5;
LP26=phase2time6;

```

Piecewise Quadratic LGM with individually-varying time scores: Empirical example code with long format input

****Including code to restructure data from long format to wide format because %PIECEWISE Macro output is in long format but Mplus LGM code expects wide format****

```

DATA: file is yourpiecewisedata.dat;
DATA LONGTOWIDE:
LONG = outcome | loadphase1 | loadphase2; !long format (original)
WIDE = y1 y2 y3 y4 y5 y6 y7 |
phase1time0 phase1time1 phase1time2 phase1time3
phase1time4 phase1time5 phase1time6 |
phase2time0 phase2time1 phase2time2
phase2time3 phase2time4 phase2time5 phase2time6; !wide format
IDVARIABLE = id;
REPETITION =occasion(0 1 2 3 4 5 6);

VARIABLE: Names are id occasion outcome
loadphase1 loadphase2; !long format (original)

Usevariables are y1-y7; !wide format

constraint= phase1time0 phase1time1
phase1time2 phase1time3 phase1time4
phase1time5 phase1time6 phase2time0
phase2time1 phase2time2 phase2time3
phase2time4 phase2time5 phase2time6; !wide format

missing=.;
ANALYSIS: estimator=ml;
MODEL:
int by y1-y7@1;
linearp1 by y1*(LP10)
y2 (LP11)
y3 (LP12)
y4 (LP13)
y5 (LP14)
y6 (LP15)
y7 (LP16);
quadp1 by y1*(QP10)
y2 (QP11)
y3 (QP12)
y4 (QP13)
y5 (QP14)
y6 (QP15)
y7 (QP16);

linearp2 by y1*(LP20)
y2 (LP21)
y3 (LP22)
y4 (LP23)
y5 (LP24)
y6 (LP25)
y7 (LP26);
quadp2 by y1*(QP20)

```

```

y2 (QP21)
y3 (QP22)
y4 (QP23)
y5 (QP24)
y6 (QP25)
y7 (QP26);

y1-y7;
[y1-y7@0];
[int linearp1 linearp2 quadp1 quadp2];
int linearp1 linearp2 quadp1 quadp2;
int linearp1 linearp2 quadp1 quadp2
with int linearp1 linearp2 quadp1 quadp2;

MODEL CONSTRAINT:
LP10=phase1time0;
LP11=phase1time1;
LP12=phase1time2;
LP13=phase1time3;
LP14=phase1time4;
LP15=phase1time5;
LP16=phase1time6;

LP20=phase2time0;
LP21=phase2time1;
LP22=phase2time2;
LP23=phase2time3;
LP24=phase2time4;
LP25=phase2time5;
LP26=phase2time6;

QP10=phase1time0*phase1time0;
QP11=phase1time1*phase1time1;
QP12=phase1time2*phase1time2;
QP13=phase1time3*phase1time3;
QP14=phase1time4*phase1time4;
QP15=phase1time5*phase1time5;
QP16=phase1time6*phase1time6;

QP20=phase2time0*phase2time0;
QP21=phase2time1*phase2time1;
QP22=phase2time2*phase2time2;
QP23=phase2time3*phase2time3;
QP24=phase2time4*phase2time4;
QP25=phase2time5*phase2time5;
QP26=phase2time6*phase2time6;

```

In the empirical example the residual variance for y_7 was very small and had to be fixed to 0 to prevent a nonpositive definite solution (see manuscript footnote); this constraint was not depicted in the above code.

Piecewise Quadratic LGM with individually-varying time scores: Empirical example code with wide format input

****This code assumes you have already taken the long format %PIECEWISE
Macro output and restructured it to wide format outside of Mplus****

```

DATA: file is yourwidepiecedata.dat;
VARIABLE:
  Names are id phase1time0-phase1time6
  phase2time0-phase2time6 y1-y7;
Usevariables are y1-y7;
constraint= phase1time0-phase1time6
phase2time0-phase2time6;
missing=.;
ANALYSIS: estimator=ml;

MODEL:
int by y1-y7@1;
linearp1 by y1*(LP10)
y2 (LP11)
y3 (LP12)
y4 (LP13)
y5 (LP14)
y6 (LP15)
y7 (LP16);
quadp1 by y1*(QP10)
y2 (QP11)
y3 (QP12)
y4 (QP13)
y5 (QP14)
y6 (QP15)
y7 (QP16);

linearp2 by y1*(LP20)
y2 (LP21)
y3 (LP22)
y4 (LP23)
y5 (LP24)
y6 (LP25)
y7 (LP26);
quadp2 by y1*(QP20)
y2 (QP21)
y3 (QP22)
y4 (QP23)
y5 (QP24)
y6 (QP25)
y7 (QP26);

y1-y7;
[y1-y7@0];
[int linearp1 linearp2 quadp1 quadp2];
int linearp1 linearp2 quadp1 quadp2;
int linearp1 linearp2 quadp1 quadp2
with int linearp1 linearp2 quadp1 quadp2;

```

Online Appendix

MODEL CONSTRAINT:

LP10=phase1time0;
LP11=phase1time1;
LP12=phase1time2;
LP13=phase1time3;
LP14=phase1time4;
LP15=phase1time5;
LP16=phase1time6;

LP20=phase2time0;
LP21=phase2time1;
LP22=phase2time2;
LP23=phase2time3;
LP24=phase2time4;
LP25=phase2time5;
LP26=phase2time6;

QP10=phase1time0*phase1time0;
QP11=phase1time1*phase1time1;
QP12=phase1time2*phase1time2;
QP13=phase1time3*phase1time3;
QP14=phase1time4*phase1time4;
QP15=phase1time5*phase1time5;
QP16=phase1time6*phase1time6;

QP20=phase2time0*phase2time0;
QP21=phase2time1*phase2time1;
QP22=phase2time2*phase2time2;
QP23=phase2time3*phase2time3;
QP24=phase2time4*phase2time4;
QP25=phase2time5*phase2time5;
QP26=phase2time6*phase2time6;

Note that in the empirical example, the residual variance for y_7 was very small and had to be fixed to 0 to prevent a nonpositive definite solution (see manuscript footnote); this constraint was not depicted in the above code.

**Structured Latent Curve for exponential LGM with individually-varying time scores:
Empirical example code**

```

DATA: file is yourdata.dat;
VARIABLE: Names are id y1-y7 time0-time6;
Usevariables are y1-y7;
constraint=time0-time6;
missing=.;
ANALYSIS: estimator=ml;
MODEL:
int by y1-y7@1 ;
totchange BY y1* (b1)
y2 (b2)
y3 (b3)
y4 (b4)
y5 (b5)
y6 (b6)
y7 (b7);
rate BY y1* (g1)
y2 (g2)
y3 (g3)
y4 (g4)
y5 (g5)
y6 (g6)
y7 (g7);
[int];
[totchange] (mutotchange);
[rate@0];
int totchange rate;
int totchange rate with int totchange rate;
y1-y7 ;
[y1-y7@0];

MODEL CONSTRAINT:
NEW (murate);
    b1 = 1-(exp(time0*(-murate)));
    b2 = 1-(exp(time1*(-murate)));
    b3 = 1-(exp(time2*(-murate)));
    b4 = 1-(exp(time3*(-murate)));
    b5 = 1-(exp(time4*(-murate)));
    b6 = 1-(exp(time5*(-murate)));
    b7 = 1-(exp(time6*(-murate)));

    g1 = mutotchange*time0*(exp(-murate*time0));
    g2 = mutotchange*time1*(exp(-murate*time1));
    g3 = mutotchange*time2*(exp(-murate*time2));
    g4 = mutotchange*time3*(exp(-murate*time3));
    g5 = mutotchange*time4*(exp(-murate*time4));
    g6 = mutotchange*time5*(exp(-murate*time5));
    g7 = mutotchange*time6*(exp(-murate*time6));

```