Ordinary Least Squares Growth Curves in R

R. Philip Chalmers

Last compiled on February 10, 2012

Introduction

Latent growth curve modeling within the structural equation modeling (SEM) framework has become a popular methodology for analyzing longitudinal data. In latent curve models (LCMs), structural variables are defined in a way that creates unobservable trajectory components that are useful for modeling how individuals or events grow over time. One of the main benefits to using LCMs is that any existing SEM software capable of estimating parameters with full-information methods (with mean structure) can be used to estimate LCMs, incidentally inheriting all of the powerful and commonly used diagnostic tools found in SEM software. While full-information estimators are widely available in commercial software, another potentially useful algorithm for estimating individual trajectories is to use ordinary least squares (OLS) estimation for each individual case, and collecting these estimates to make inferences about the latent trajectory parameters. This technique is available in any statistical software that is capable of estimating ordinary linear regression models, and a user defined macro has already been written for SAS software by Carrig, Wirth, and Curran (2004). In this article, we shall demonstrate how similar results can be obtained in R (R Development Core Team, 2011).

As Bollen and Curran (2006) note, there are a variety of advantages to using the case-by-case approach for estimating trajectory parameters. First of all, OLS estimation is intuitively appealing, making it a good pedagogical tool for introducing how to model trajectories, and illuminates many essential conditions and assumptions necessary for LCMs. Second, prediction of the parameters for individual trajectory estimates are calculated for each case in the sample, which can lead to several diagnostics by statistical and graphical means. Also, summary statistics can be computed for these estimates (which can also be graphically portrayed) and if need be these estimates can be analyzed further by other statistical frameworks.

Unfortunately there are also several limitation to OLS estimation for LCMs, namely: overall tests of fit are not readily available, the structure of the error variances must be unrealistically constrained to estimate a pooled standard error, the latent factors cannot be regressed without error on other exogenous or time-varying variables, and analytic significance tests are often not readily available (Bollen & Curran, 2006). However, OLS estimation may still be useful in the preliminary stages of latent curve modeling for (a) selecting appropriate functional forms of growth, (b) examining unconditional population homogeneity, (c) observing whether the relationship between growth factors are linear, and for (d) detecting influential outliers (Carrig et al., 2004; Rogosa, 1994).

Parameter Estimation of OLS Curve Models

OLS estimation begins by estimating N OLS regression models, one for each case in the sample. In the case of a linear functional form of growth, the model estimated for each case is

$$y_{it} = \alpha_i + \lambda_t \beta_i + \epsilon_{it} \tag{1}$$

with

$$\alpha_i = \mu_\alpha + \zeta_{\alpha i} \tag{2}$$

$$\beta_i = \mu_\beta + \zeta_{\beta i} \tag{3}$$

where y_{it} is the value of the repeated measure for individual i at time t, α_i is the regression intercept for individual i, λ_t is the value of the user specified coding of time at time t, β_i describes linear change over time in y for individual i, and ϵ_{it} is the time-specific regression residual for individual i. The OLS estimator of slope for each case reduces to

$$\hat{\beta}_i = \frac{\sum_{t=1}^{T}(\lambda_t - \bar{\lambda})(y_{it} - \bar{y}_i)}{\sum_{t=1}^{T}(\lambda_t - \bar{\lambda})^2} \tag{4}$$

and the OLS estimator of intercept for each case reduces to

$$\hat{\alpha}_i = \bar{y}_{it} - \hat{\beta}_i\bar{\lambda} \tag{5}$$

Standard errors for any group mean ($\hat{\mu}_\psi$) are available and can be calculated as

$$\text{s.e.}(\hat{\mu}_\psi) = \sqrt{\frac{\sum(\hat{\psi}_i - \hat{\mu}_\psi)^2/(N-1)}{N}} \tag{6}$$

These estimates may also be useful in hypothesis testing situations for determining which mean trajectory parameters would be useful to model.

The variances of the trajectory parameters can be calculated using the sample variance estimates of the individual trajectories, however these values are biased representations of the population variance parameters. Though formulae for the correction of linear trajectory parameters do exist (see Rogosa, 1994) they are not implemented in any `OLScurve.R` functions. However, users can still make use of the unadjusted variance estimates by treating them as *upper bounds* of the true variance parameters. This interpretation is possible because the calculation of the variances from the trajectory parameters does not account for errors in the estimation at the case-by-case level, but if case-by-case errors were known, or could be approximated, then they would simply remove a portion of variance due to the inherent error in predicting the individual trajectories (see Bollen & Curran, 2006, p. 28). Hence accounting for the case level errors only serves to (potentially) lower the group variance estimates.

OLScurve in R

The primary function for creating objects from the `OLScurve` package is, perhaps unsurprisingly, the `OLScurve()` function, which requires two basic inputs: a formula with the keywords *y* for the DV and *time* for the time dependent functional form of growth, and a data-set containing only the unconditional growth variables with the individual trajectories in the rows. `OLScurve()` contains an additional input parameter, `time`, which specifies the time difference in the sequentially recorded occasions, where by default the variables are assumed to be equally spaced (e.g., `time = 0,1,2,3,...,T-1`).

The `OLScurve` functions also subtly improves upon various areas in the SAS macro created by Carrig et al. (2004). The first feature added is that `OLScurve()` can model a wide variety of growth functional forms, whereas the SAS macro can only model linear and quadratic growth. For example, consider the following `formula` specified functional forms, which when combined with the ability for the user to specify the time metric allows for flexible modeling of individual trajectories:

- Linear → ~ `time`
- Cubic Polynomial → ~ `time + I(time^2) + I(time^3)`

- Square-root → ~ `sqrt(time)`
- Exponential → ~ `exp(time)`
- Combination → ~ `time + I(time^2) + sqrt(time)`

The next improvement can be found in the `print(object)` function, which produces summary statistics for the entire data or by groups to observe sample homogeneity. Here the pooled standard error estimate is printed, as well as the standard errors for the mean trajectories and covariances for the entire data, potentially partitioned into groups.

All of the graphical capabilities from the SAS macro have been ported to R with some modifications to better represent the data when a conditional grouping variable is of interest, and new graphical features have been added to allow further probing of the case-by-case trajectory estimates. These feature capitalize on the powerful and flexible trellis based graphical package `lattice` (Sakar, 2008). To begin, the generic `plot(OLSobject)` function produces the individual trajectories for each grouping variable onto a single plot, useful for comparing the homogeneity of slopes and intercepts between groups. The `subjplot(OLSobject)` function creates a grid of the case-by-case trajectories superimposed over the raw data, and this is useful to determine visually how well each trajectory predicts each case. Finally, `parplot(OLSobject)` produces graphics for observing the estimated parameters values, displaying either histograms, box-plots, or scatter-plot matrices.

Nonlinear Factor Relationship Example

A short example of the functions available in `OLScurve.R` is presented here that demonstrates how to detect nonlinear relationships between latent factors. A data set entitled `nonlindata` consisting of $N = 500$ trajectories with four distinct and equally spaced time-dependent variables was simulated from the parameters $\mu_\alpha = 0$, $\text{VAR}(\alpha) = 1$, $\mu_\beta = 1$, $\text{VAR}(\beta) = 0.5$, $\zeta_\alpha = \zeta_\beta = \epsilon_i = N \sim (0, 0.2)$. However, the relationship between the α and β parameters was specified to be that of a U-shaped relationship; specifically, $\beta = z_\alpha \sqrt{\text{VAR}(\beta)} + \mu_\beta$, where z_α represents the z-score standardization of $|\alpha|^2$. For didactic reasons, a hypothetical grouping variable 'gender' was created to demonstrate how the functions can accommodate nominal grouping variables.

Although we should suspect that a linear growth model would be best for these data, a quadratic relationship is tested first.

```
> library("OLScurve")
> mod.quad <- OLScurve( ~ time + I(time^2), data = nonlindata)
> mod.quad

Call:
OLScurve(formula = ~time + I(time^2), data = nonlindata)
Note: 0% ommited cases.
Pooled standard error =  0.01989803

MEANS:

$fulldata
(intercept)        time      time^2
     -0.018       1.000      -0.003

Standard Errors for Means:
```

```
$fulldata
(intercept)        time      time^2
      0.053       0.051       0.026


COVARIANCE (correlations on lower off-diagonal):

$fulldata
            (intercept)   time time^2
(intercept)       1.056  0.018  0.004
time              0.022  0.651 -0.029
time^2            0.036 -0.365  0.010
```

We observe from the output that the mean trajectory for the quadratic factor is not of great magnitude, and in fact fails to reach significance ($z \approx -0.003/0.026 = 0.115$, $p = .54$). We also observe that there is relatively little variance in the quadratic factor, so the conclusion not use a quadratic term appears to be corroborated. We now compose a linear trajectory model, determine if there are any major differences between the grouping variables, and observe how well this model fits the data at the individual trajectory level.

```
> mod.lin <- OLScurve( ~ time, data = nonlindata)
> subjplot(mod.lin)

> print(mod.lin, group = gender)

Call:
OLScurve(formula = y ~ time, data = nonlindata)
Note: 0% ommited cases.
Pooled standard error =  0.03981435

MEANS:

$male
(intercept)        time
     -0.014       0.984

$female
(intercept)        time
     -0.015       0.999

Standard Errors for Means:

$male
(intercept)        time
      0.079       0.061

$female
(intercept)        time
```

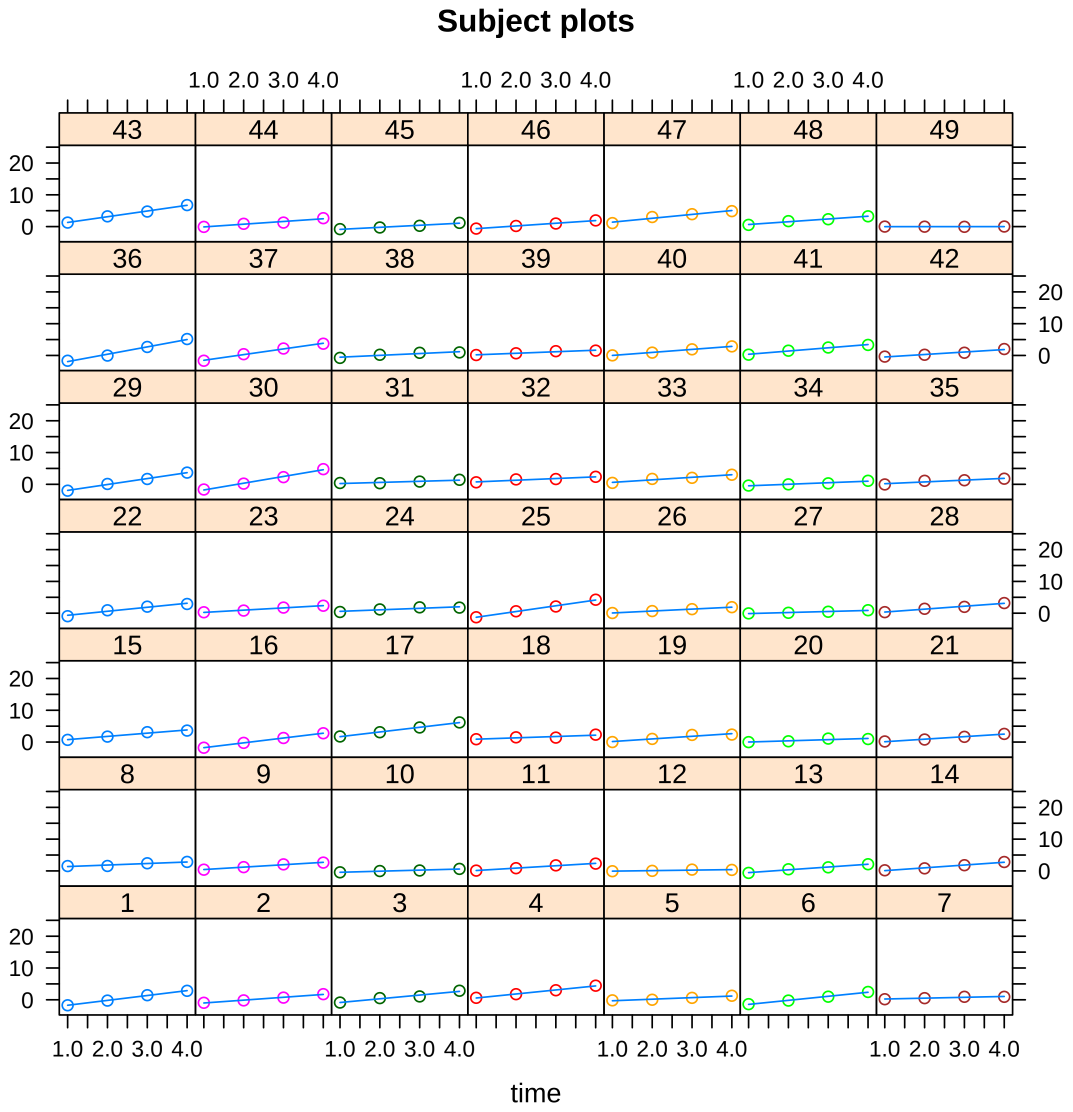

Figure 1. First 49 cases from `subjplot()`.

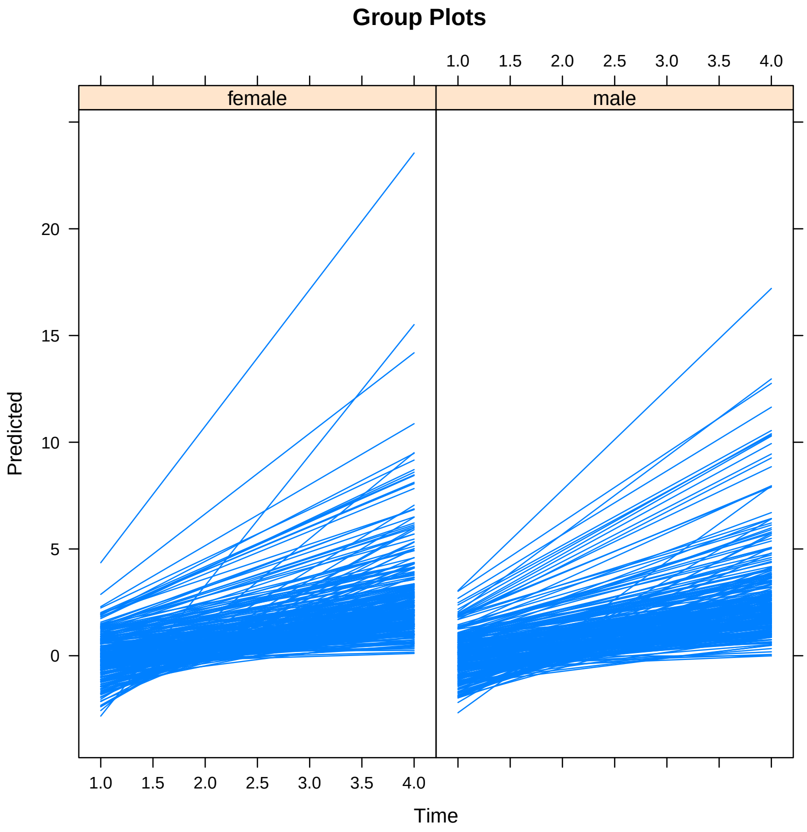

Figure 2. Group allocated plot of the modeled trajectories.

```
      0.081       0.068


COVARIANCE (correlations on lower off-diagonal):

$male
            (intercept)  time
(intercept)       1.036 0.100
time              0.142 0.478

$female
            (intercept)   time
(intercept)       1.085 -0.042
time             -0.050  0.652
```

```
> plot(mod.lin, group = gender)
```

Next, we use a scatter-plot matrix graphic (Figure 3) to determine whether the assumption of linearity between trajectory parameters is tenable. Although comparing trajectory parameter relationships does not appear to be common practice according to texts on latent curve modeling (e.g., Bollen & Curran, 2006) it is nonetheless an important area that should be probed to ensure that a proper LCM model has been fit to the data. For example, if one were to use Mplus (Muthén & Muthén, 2008) to fit a linear LCM to this simulated data they would find that the overall model fit extremely well, $\chi^2(5) = 3.652$, $p = .60$, where the correlation between the slope and intercept factors is $r_{\alpha\beta} = 0.053$, $z = 1.150$, $p = .25$. However, graphical inspection of the OLS parameters using `parplot()` reveals that even though the overall model fit may be excellent the conclusion that the intercept and slope are unrelated is far from correct (although it is true that there is no *linear* relationship). It appears that individuals with higher *and* lower initial intercepts tend to have larger slope trajectories, whereas those closely centered around the intercept mean have much smaller slopes. Similar conclusions could be drawn from estimating factor scores from the LCM in Mplus or other software, although the choice of which estimation method to employ may make a difference, and unfortunately researchers are often not comfortable analyzing calculated factor scores—perhaps due to the large literature surrounding factor score indeterminacy (Bollen, 1989; Mulaik, 2010).

```
> parplot(mod.lin, type = 'splom')
```

Discussion

As was demonstrated above, R now has the ability to comfortably implement OLS growth curves with the help of the `OLScurve`, and improves upon Carrig et al.'s (2004) SAS macro. Plotting the case-by-case trajectories has been substantially condensed to allow for viewing multiple cases per graph, compared to the SAS macro which plots one case per graphic, and other high quality graphics have been made available from the `lattice` package to facilitate analysis. The flexibility of model specification has also been greatly improved, since `OLScurve()` is not limited to modeling only linear and quadratic growth curves. Additionally the object returned by `OLScurve()` may have its individual elements extracted for further analyses that were not contained in the source code. Finally, and for completeness, selections of cases (via row extraction) defined by the user to allow sub-groupings to be analyzed separately is possible by simply extracting which rows in the `data` input are to be grouped and placing them in separate `matrix` or `data.frame` objects.

In conclusion, OLS estimation can be used to evaluate many important and often overlooked model assumptions. Visually presenting estimates of individual growth trajectories allows for the identification of cases for which the selected functional form of growth may not characterize the manifest data, and further examination of these cases could reveal that these cases are indeed outliers. In this way, OLS estimation can be used either as a screening tool or as a post-hoc diagnostic tool to assess model misfit (Carrig et al., 2004). We hope that the range of flexibility offered by the `OLScurve` functions prove useful for researchers—who may or may not be regular R users—interested in broadening their latent curve modeling toolbox.

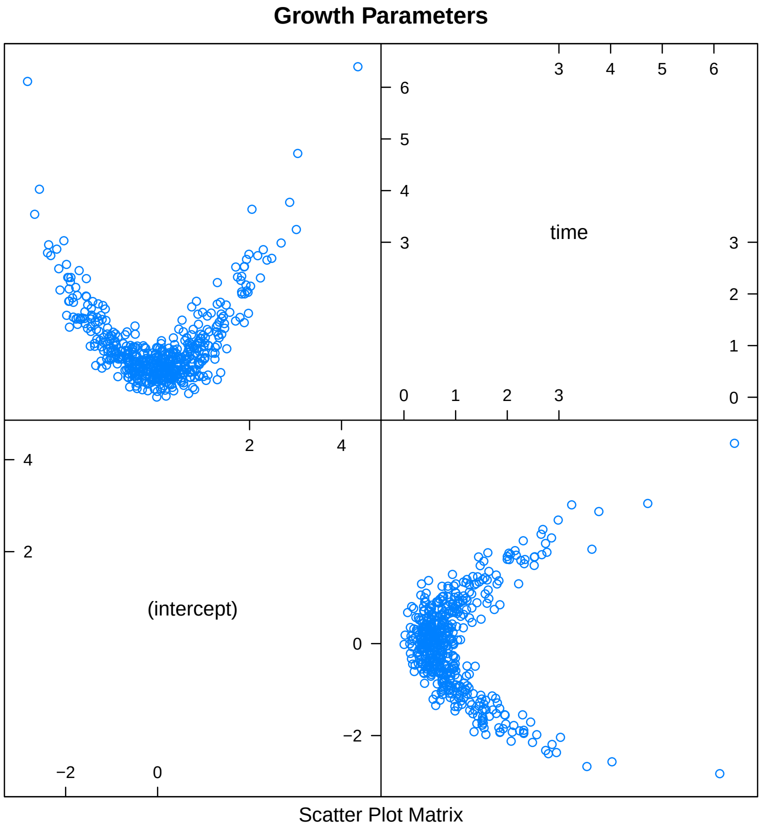

Figure 3. Scatter-plot matrix between the intercept and slope parameter estimates.

References

Bollen, K. A. (1989). *Structural equations with latent variables*. New York: John Wiley.

Bollen, K. A., & Curran, P. J. (2006). *Latent curve models: A structural equation perspective*. Hoboken, NJ: John Wiley & Sons.

Carrig, M. M., Wirth, R. J., & Curran, P. J. (2004). A SAS macro for estimating and visualizing individual growth curves. *Structural Equation Modeling*, *11* (1), 132–149.

Mulaik, S. A. (2010). *Foundations of factor analysis* (2nd ed.). Boca Raton, FL: Chapman & Hall.

Muthén, L. K., & Muthén, B. O. (2008). Mplus (Version 5.0) [Computer Program]. Author.

R Development Core Team. (2011). *R: A Language and Environment for Statistical Computing*. ISBN 3-900051-07-0. R Foundation for Statistical Computing. Vienna, Austria. Retrieved from R Foundation for Statistical Computing: `http://www.R-project.org/`

Rogosa, D. R. (1994). Individual trajectories as the starting point for longitudinal data analysis. *Alzheimer Disease and Associated Disorders*, *8*, 302–307.

Sakar, D. (2008). *Lattice: Multivariate data visualization with R*. New York, NY: Springer.