Fitting time-varying effects models

John J. Dziak

2020-06-25

In this example we simulate a longitudinal dataset and fit a simple time-varying coefficient model to it using the tvem package. We show that this can be done for either a continuous or a binary outcome variable. Time-varying coefficient models are discussed further by Tan, Shiyko, Li, Li and Dierker (2012) and are an application of the varying-coefficient models approach of Hastie and Tibshirani (1993) to intensive longitudinal data.

Before running the examples, first load the tvem package.

```
library(tvem)
#> Loading required package: mgcv
#> Loading required package: nlme
#> This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.
```

Example with a continuous outcome variable

The tvem library has a function for simulating a dataset. It is good to start by specifying a random seed.

```
set.seed(123);
the_data <- simulate_tvem_example();
```

Exploring the dataset

When analyzing any dataset, it is important to examine it descriptively first.

```
print(head(the_data));
#>   subject_id time  x1  x2   y
#> 1          1 0.00 5.3 5.8 0.9
#> 2          1 0.05 6.4 5.3 0.0
#> 3          1 0.10 5.4 5.5  NA
#> 4          1 0.15 5.7 3.2  NA
#> 5          1 0.20 5.8 2.8  NA
#> 6          1 0.25 8.7 3.2 1.9
print(summary(the_data));
#>    subject_id          time            x1              x2
#>  Min.   :  1.00   Min.   :0.00   Min.   : 0.00   Min.   : 0.000
#>  1st Qu.: 75.75   1st Qu.:1.75   1st Qu.: 3.60   1st Qu.: 1.900
#>  Median :150.50   Median :3.50   Median : 5.00   Median : 3.300
#>  Mean   :150.50   Mean   :3.50   Mean   : 5.02   Mean   : 3.326
#>  3rd Qu.:225.25   3rd Qu.:5.25   3rd Qu.: 6.40   3rd Qu.: 4.700
#>  Max.   :300.00   Max.   :7.00   Max.   :10.00   Max.   :10.000
#>
#>        y
#>  Min.   :0.000
```

```
#>  1st Qu.:1.000
#>  Median :2.200
#>  Mean   :2.308
#>  3rd Qu.:3.400
#>  Max.   :9.600
#>  NA's   :29647
```

The dataset is in long form (one row per observation, with multiple observation times for each participant). There are 300 participants. The observation time ranges from 0 to 7, imagined as seven days. There is a single response variable y, and two predictor variables (covariates), x_1 and x_2. In the context of an intensive longitudinal study with human participants, these variables might be ratings of different feelings, symptoms or behaviors, reported a few times per day at random times, during seven days following an event (such as the beginning of an intervention, treatment, lifestyle change, etc.). The values of the covariates and the response vary over time within subject. The analyst wishes to find out whether their means change systematically over time, whether they are interrelated, and whether this relationship, if it exists, changes over time.

Plotting average change over time

One easy thing to do is to investigate whether and how the response changes over time on average. This is simply curve fitting, similar to polynomial regression, but can be fit using the TVEM function, in an approach sometimes called 'intercept-only TVEM.' This approach uses a spline function to approximate the average change in y over time.

By default, the tvem function will fit a penalized B-spline (de Boor, 1972), which is called a "P-spline" in the terminology of Eilers and Marx (1996). This approach uses an automatic tuning penalty to choose the level of smoothness versus flexibility of the fitted function. It is similar, though not identical, to the P-splines used in the Methodology Center's %TVEM SAS macro, which are penalized truncated power splines (Li et al., 2017; see Ruppert, Wand & Carroll, 2003).

```
model1 <- tvem(data=the_data,
               formula=y~1,
               id=subject_id,
               time=time);
```

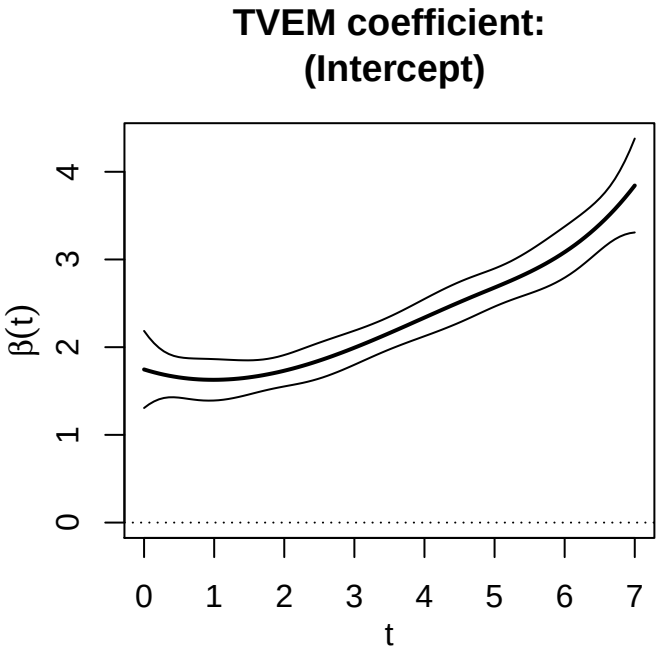
You also have the option to turn off the penalty and control the smoothness yourself, by specifying the number of interior knots, here 2.

```
model1 <- tvem(data=the_data,
               formula=y~1,
               id=subject_id,
               num_knots=2,
               penalize=FALSE,
               time=time);
```

The implied mean model is $E(y|t) = \beta_0(t)$ Where t is time in days. After fitting the model, you can print a summary and plot the estimated coefficient.

```
print(model1);
#> ====================================================
#> Time-Varying Effects Modeling (TVEM) Function Output
#> ====================================================
#> Response variable:   y
#> Time interval:   0 to 7
#> Number of subjects:  300
#> Effects specified as time-varying:  (Intercept)
#> You can use the plot_tvem function to view their plots.
```

```
#> ======================================================
#> Back-end model fitted in mgcv::bam function:
#> Method REML
#> Formula:
#> y ~ s(time, bs = "ps", by = NA, pc = 0, k = 6, fx = TRUE)
#> Pseudolikelihood AIC: 46234.19
#> Pseudolikelihood BIC: 46260.11
#> Note: Used listwise deletion for missing data.
#> ======================================================
plot(model1);
```



The plot shows the estimated coefficient function, and approximate estimates for 95% pointwise confidence intervals (not corrected for potential multiple comparisons) for the value of the function at each time.

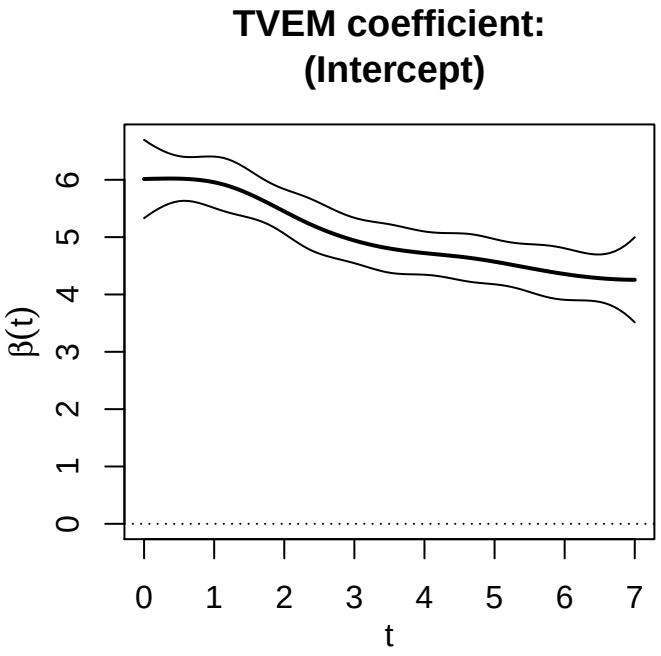
We don't provide for random effects in the current version of this package. Instead, we use a (possibly penalized) form of generalized estimating equations with working independence, and adjust the standard errors for within-subject correlation using a sandwich formula.

Using the select_tvem function

It is a very good idea to examine how the covariate means change over time. That is, we should fit intercept-only TVEM's for x_1 and x_2, not just y. While doing this, we can take the opportunity to additionally explore yet another way to fit a model using the tvem function, by using the select_tvem function to choose the number of knots by an pseudolikelihood equivalent to an AIC or BIC criterion. "Pseudolikelihood" here means that the information criterion doesn't take within-subject correlation into account, because we are trying to fit a marginal model agnostic to the exact correlation structure.

```
model2 <- select_tvem(data=the_data,
                      formula = x1~1,
                      id=subject_id,
```

```
                    time=time);
#>      knots       ic
#> [1,]     0 177522.9
#> [2,]     1 177475.0
#> [3,]     2 177453.0
#> [4,]     3 177452.0
#> [5,]     4 177448.3
#> [6,]     5 177446.4
#> [1] "Selected 5 interior knots."
print(model2);
#> ======================================================
#> Time-Varying Effects Modeling (TVEM) Function Output
#> ======================================================
#> Response variable:   x1
#> Time interval:   0 to 7
#> Number of subjects:  300
#> Effects specified as time-varying:  (Intercept)
#> You can use the plot_tvem function to view their plots.
#> ======================================================
#> Back-end model fitted in mgcv::bam function:
#> Method fREML
#> Formula:
#> x1 ~ s(time, bs = "ps", by = NA, pc = 0, k = 9, fx = FALSE)
#> Pseudolikelihood AIC: 177446.42
#> Pseudolikelihood BIC: 177476.67
#> Model selection table for number of interior knots:
#>      knots       ic
#> [1,]     0 177522.9
#> [2,]     1 177475.0
#> [3,]     2 177453.0
#> [4,]     3 177452.0
#> [5,]     4 177448.3
#> [6,]     5 177446.4
#> ======================================================
plot(model2);
```



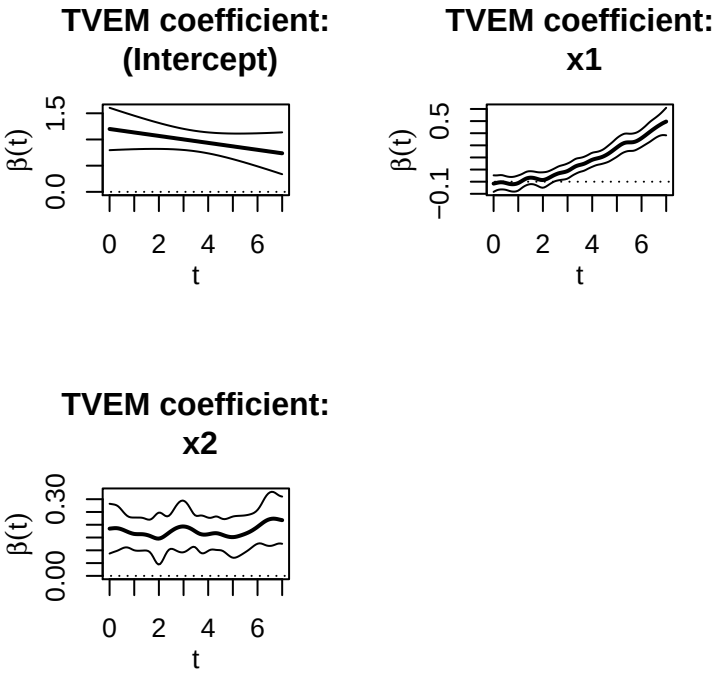
It would be good to plot the mean of x_2 over time also in the same way.

Time-varying effects of covariates

After this and further exploration of the data, we can go ahead to fit a nontrivial TVEM model, with covariates. We allow both x_1 and x_2 to potentially have "time-varying effects" (regression relationships with the response that change over time, that is, a potential interaction between time and the covariate, specified without the assumption of linearity). The implied mean model is $E(y|t, x_1(t), x_2(t)) = \beta_0(t) + \beta_1(t)x_1(t) + \beta_2(t)x_2(t)$ where t is time in days.

```
model3 <- tvem(data=the_data,
               formula=y~x1+x2,
               id=subject_id,
               time=time);
print(model3);
#> ========================================================
#> Time-Varying Effects Modeling (TVEM) Function Output
#> ========================================================
#> Response variable:   y
#> Time interval:   0 to 7
#> Number of subjects:  300
#> Effects specified as time-varying:  (Intercept), x1, x2
#> You can use the plot_tvem function to view their plots.
#> ========================================================
#> Back-end model fitted in mgcv::bam function:
#> Method fREML
#> Formula:
#> y ~ x1 + x2 + s(time, bs = "ps", by = NA, pc = 0, k = 24, fx = FALSE) +
#>     s(time, bs = "ps", by = x1, pc = 0, m = c(2, 1), k = 24,
#>         fx = FALSE) + s(time, bs = "ps", by = x2, pc = 0, m = c(2,
```

```
#>     1), k = 24, fx = FALSE)
#> Pseudolikelihood AIC: 44272.03
#> Pseudolikelihood BIC: 44373.9
#> Note: Used listwise deletion for missing data.
#> =====================================================
plot(model3);
```



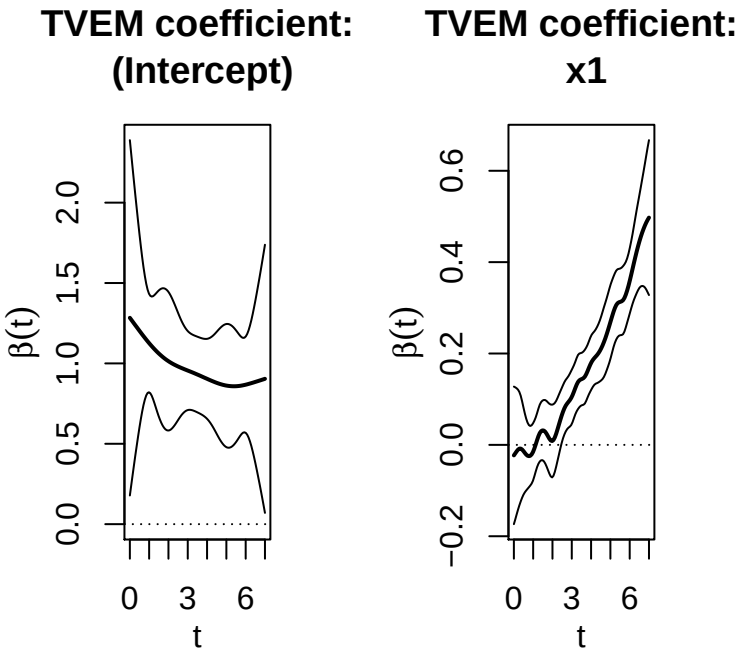
Holding x_1 and x_2 constant, the mean of y seems to decline over time. The penalty function causes the the relationship to be estimated as linear because there is no clear evidence of nonlinearity. From the results, x_1 appears to have an increasingly positive relationship with y over time. x_2 also seems to predict y, but the strength of the relationship does not change over time. Thus, we could reasonably fit a similar but simpler model, but with x_2 having a non-time-varying effect even though it has time-varying values.

Time-invariant effects of covariates

The following code fits a model where x_2 is assumed to have a time-invariant effect but x_1 is allowed to have a time-varying effect.

```
model4 <- tvem(data=the_data,
               formula=y~x1,
               invar_effect=~x2,
               id=subject_id,
               time=time);
print(model4);
#> =====================================================
#> Time-Varying Effects Modeling (TVEM) Function Output
#> =====================================================
#> Response variable:   y
#> Time interval:   0 to 7
#> Number of subjects:  300
```

```
#> Effects specified as time-varying:  (Intercept), x1
#> You can use the plot_tvem function to view their plots.
#> ========================================================
#> Effects specified as non-time-varying:
#>     estimate standard_error
#> x2 0.1753651     0.01585862
#> ========================================================
#> Back-end model fitted in mgcv::bam function:
#> Method fREML
#> Formula:
#> y ~ x1 + x2 + s(time, bs = "ps", by = NA, pc = 0, k = 24, fx = FALSE) +
#>     s(time, bs = "ps", by = x1, pc = 0, m = c(2, 1), k = 24,
#>         fx = FALSE)
#> Pseudolikelihood AIC: 44285.93
#> Pseudolikelihood BIC: 44370.6
#> Note: Used listwise deletion for missing data.
#> ========================================================
plot(model4);
```



The implied mean model is $E(y|t) = \beta_0(t) + \beta_1(t)x_1(t) + \beta_2 x_2(t)$. Notice that only covariates with time-varying effects are listed in the formula argument. The invar_effect argument is used for covariates with time-invariant effects. Also notice that a tilde ( ) sign is needed before the list of covariates with time-invariant effects, because it is treated by R as the right side of a formula. If there were multiple covariates with time-invariant effects, they would be listed as follows: "~x2+x3+x4."

In the output, there is no longer a plot for the coefficient of x_2 as a function of time, because β_2 is considered constant over time even if x_2 varies. Instead, the estimate and estimated standard error of β_2 are given as text in the output from print_tvem.

The seeming oscillations in confidence interval width in the plot do not have any particular interpretation;

they are just an artifact of the locations of the knots. Also, the confidence intervals shown are approximate pointwise confidence intervals, much like those in the Methodology Center's %TVEM SAS macro. They are not simultaneous confidence bands, so they do not directly correct for multiple comparisons.

The main takeaway message from the analysis is the increasing $\beta_1(t)$ over time, suggesting an increasing association between x_1 and y, that is, some kind of interaction between t and x_1 in predicting y.

Example with a binary outcome variable

This example will be similar to the previous one, but with a binary response variable. The tvem library's simulation function will also simulate binary y if requested by an optional argument.

```
set.seed(123);
the_data <- simulate_tvem_example(simulate_binary=TRUE);
```

The simulated dataset is similar to the previous example, but with binary y (0=no, 1=yes) generated from a logistic model.

Plotting the average log odds over time

We can plot the expected log odds over time, using an time-varying-intercept-only logistic model. The model assumes $\text{logit}(E(Y|t)) = \beta_0(t)$. The binary outcome is specified using the family argument as in R's glm function.

```
model1 <- tvem(data=the_data,
               formula=y~1,
               family=binomial(),
               id=subject_id,
               time=time);
```

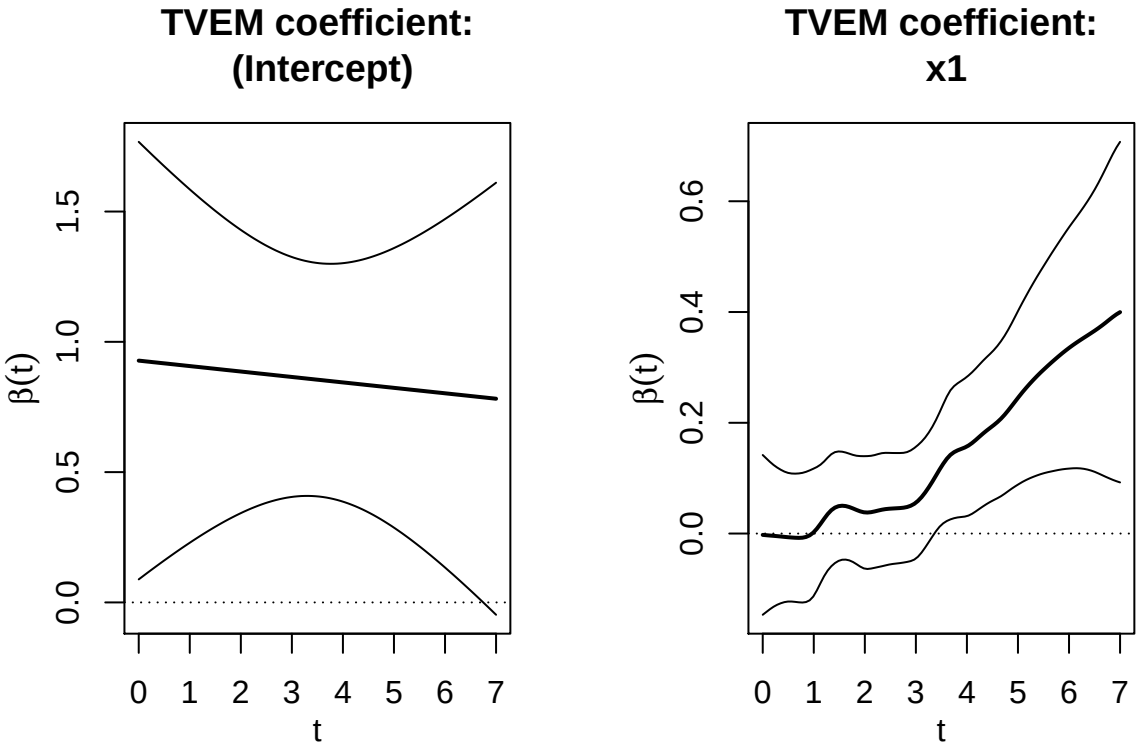
As before, you can use the default option of an automatic penalty function, or you could specify the number of knots, or you could use automatic selection for the number of knots without a penalty.

Including covariates

A model with covariates works similarly to the previous example also.

```
model2 <- tvem(data=the_data,
               formula=y~x1,
               invar_effect=~x2,
               id=subject_id,
               family=binomial(),
               time=time);
print(model2);
#> ========================================================
#> Time-Varying Effects Modeling (TVEM) Function Output
#> ========================================================
#> Response variable:   y
#> Time interval:   0 to 7
#> Number of subjects:  300
#> Effects specified as time-varying:  (Intercept), x1
#> You can use the plot_tvem function to view their plots.
#> ========================================================
#> Effects specified as non-time-varying:
#>     estimate standard_error
#> x2 0.2074037     0.04819109
```

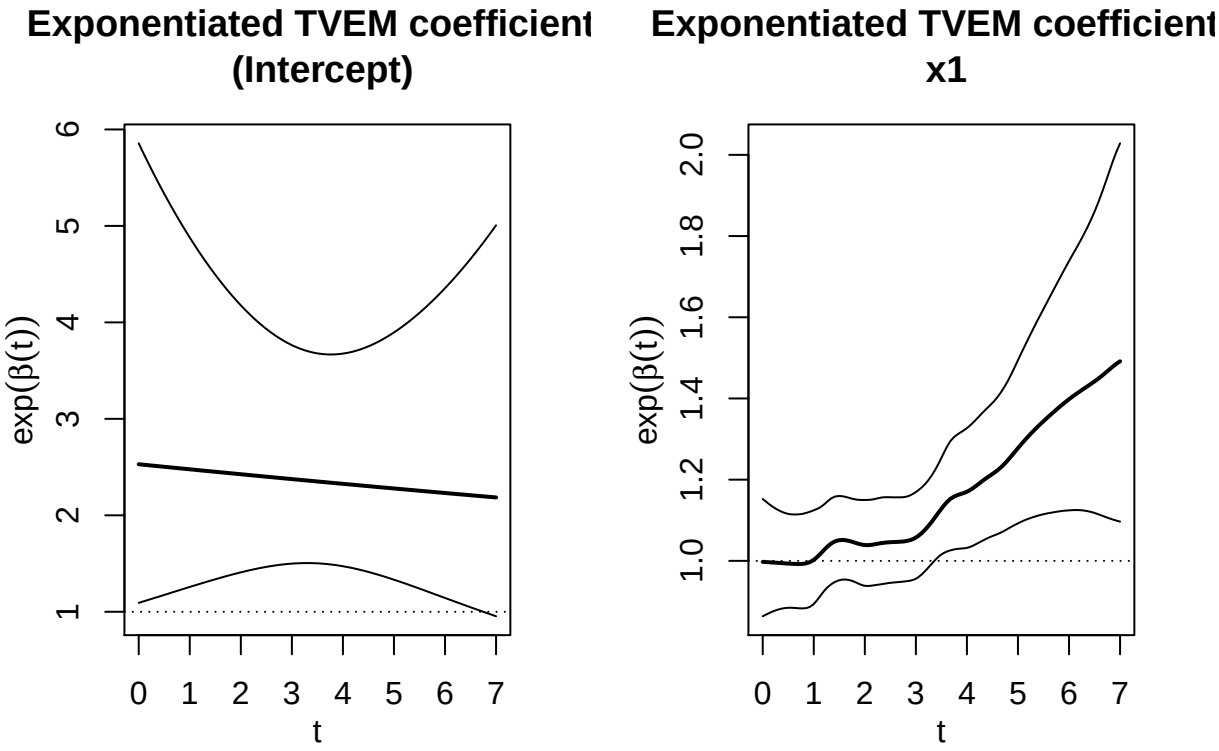
```
#> ======================================================
#> Back-end model fitted in mgcv::bam function:
#> Method fREML
#> Formula:
#> y ~ x1 + x2 + s(time, bs = "ps", by = NA, pc = 0, k = 24, fx = FALSE) +
#>     s(time, bs = "ps", by = x1, pc = 0, m = c(2, 1), k = 24,
#>         fx = FALSE)
#> Pseudolikelihood AIC: 8609.09
#> Pseudolikelihood BIC: 8657.7
#> Note: Used listwise deletion for missing data.
#> ======================================================
plot(model2);
```



The implied mean model is $\text{logit}(E(y|t)) = \beta_0(t) + \beta_1(t)x_1(t) + \beta_2 x_2(t)$. In this simulated example, it isn't very clear whether the effect of x_1 changes over time or not, although it seems to do so. Logistic regression analyses often have wider confidence intervals than ordinary regression analyses with the same sample size.

With logistic TVEM, we can additionally plot exponentiated coefficients to get odds and odds ratios. In the plots below, the exponentiated intercept function shows the estimated odds of $Y = 1$ at a given time, assuming $x_1 = x_2 = 0$. The exponentiated beta function for x_1 shows the estimated odds ratio for $Y = 1$ given a 1-unit increase in x_1 at a given time.

```
plot(model2, exponentiate=TRUE);
```



Binary outcomes are not the only non-normal outcomes which can be modeled using TVEM. For example, Poisson count outcomes with a log link can be modeled by specifying family=poisson() instead of family=binomial(). However, for overdispersed or zero-inflated counts, the Poisson distribution might not be appropriate.
We also do not currently have a simulation function in the tvem package for Poisson data, because generating realistic longitudinal count data is beyond the scope of this package.

References

- de Boor, C. (1972). On calculating with B-splines. Journal of Approximation Theory, 6: 50–62.
- Eilers, P. H. C., & Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. Statistical Science, 11: 89-121.
- Hastie, T, Tibshirani, R. (1993). Varying-coefficient models. Journal of the Royal Statistical Socety, B, 55:757-796.
- Li, R., Dziak, J. J., Tan, X., Huang, L., Wagner, A. T., & Yang, J. (2017). TVEM (time-varying effect model) SAS macro users' guide (Version 3.1.1). University Park: The Methodology Center, Penn State.
- Ruppert, D., Wand, M. P., & Carroll, R. J. (2003). Semiparametric regression. Cambridge: Cambridge University Press.
- Tan, X., Shiyko, M. P., Li, R., Li, Y., & Dierker, L. (2012). A time-varying effect model for intensive longitudinal data. Psychological Methods, 17: 61-77.