Fitting a tvem with binary output

John J. Dziak

2020-06-05

This example will be similar to the previous one, but with a binary response variable.

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

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);
```

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 NA
#> 2          1 0.05 6.4 5.3 NA
#> 3          1 0.10 5.4 5.5 NA
#> 4          1 0.15 5.7 3.2  1
#> 5          1 0.20 5.8 2.8 NA
#> 6          1 0.25 8.7 3.2 NA
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 :1.000
#>  Mean   :0.949
#>  3rd Qu.:1.000
#>  Max.   :1.000
#>  NA's   :29647
```

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

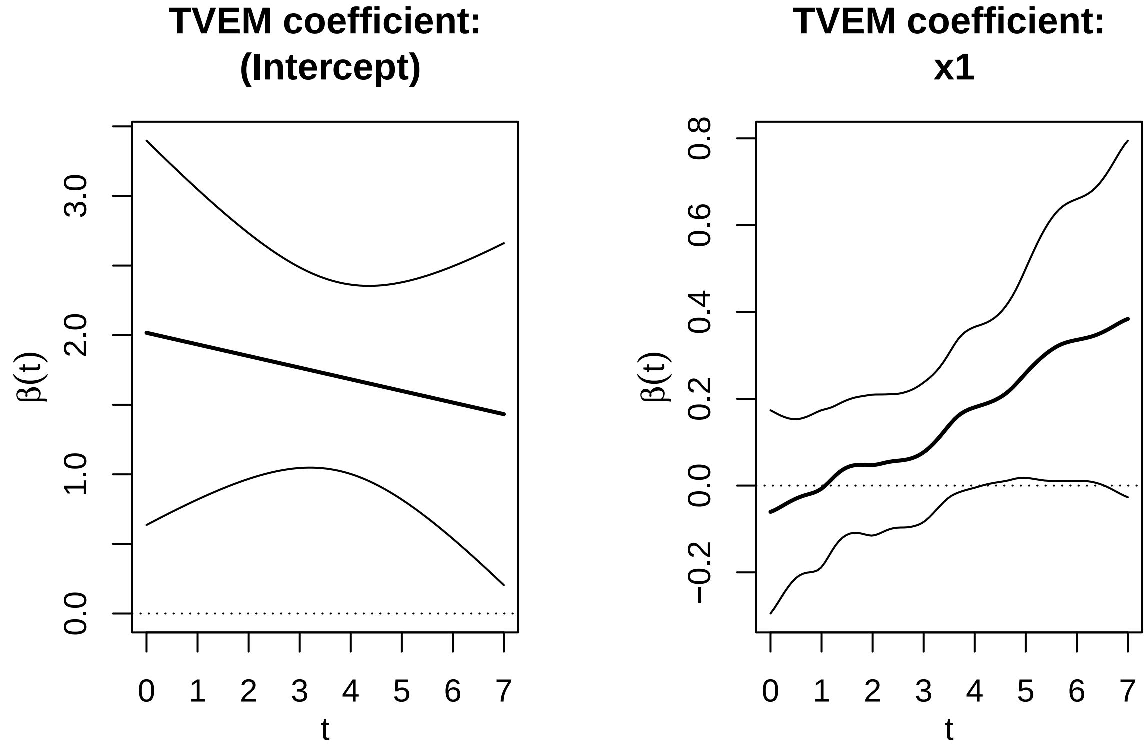
We can plot the expected log odds over time, using an time-varying-intercept-only logistic model. The model assumes $\text{logit}(E(Y|t)) = \beta_0(t)$. The binary outcome is specified using the family argument as in R's glm function.

```
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.

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 0.224529      0.0765376
#> =======================================================
#> 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: 4820.89
#> Pseudolikelihood BIC: 4863.41
#> 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.