An Example of Data Analysis Using reVAR

This report provides the steps to generate results similar to Table 3,4 and Figure 2 of Sun et al. (2020+) using a simulated dataset.

```
library(reVAR)
library(survival)
library(ggplot2)
library(knitr)
```

Data preparation

The function `genData()` can be used to generate the simulated dataset. The ten possible data generating mechanisms in `genData()` correspond to Scenarios I-X in Sun et al. (2020+), Section 4.

```
set.seed(123456)
dat <- genData(200, sce = "VAR3")
```

To use the function `reVAR()`, three data frames need to be prepared:

1. **nonEventDF1**: A data frame that records all the information for fitting the non-event visit model. Specifically, the first four columns must be `ID`, `Start`, `End`, `Status`:

- `ID`: The subject ID for each observation. One subject can have multiple rows;
- `Start`: The starting time for the interval;
- `End`: The ending time of the interval;
- `Status`: The status indicator, 1 = non-event visit, 0 = no;
- The other columns are covariate values for the non-event model on the time interval `(Start, End]`.

In this example, there are three covariates (i.e., X1(t), X2(t), X3(t)) in the non-event model. **The covariates for the non-event visit model are continuously observed during the follow-up period**. The function `genData()` generates a data frame `nonEventDF1_s2`, which contains the above information and an additional column `Status2` (1 = event visit, 0 = no). The variable `Status2` needs to be removed when using `reVAR()`; `Status2` will be used later when fitting the model using the last-covariate-carrying-forward (LCCF) approach.

```
nonEventDF1 <- subset(dat$nonEventDF1_s2, select = -Status2)
head(nonEventDF1)
```

```
##   ID  Start      End Status X1         X2         X3
## 1  1 0.0000 1.187000      1  0 -0.3792345 0.06562064
## 2  1 1.1870 2.141500      1  1  0.8285129 0.06562064
## 3  1 2.1415 3.988922      0  1 -0.7402917 0.06562064
## 4  2 0.0000 0.181000      1  0  0.9967859 0.21085317
## 5  2 0.1810 0.863500      1  0  0.8830705 0.21085317
## 6  2 0.8635 1.014500      1  0 -0.8732199 0.21085317
```

The other two data frames record the event visit information. The covariates in the event model (i.e., Z1(t), Z2(t), Z3(t)) are only observed at events and non-event visits.

2. **nonEventDF2**: A data frame that records covariates for the event model **at non-event visits**. The first two columns must be ID and Time:

- `ID`: The subject ID for each observation;
- `Time`: Time zero and the non-event visit times for all the subjects;
- The other columns are covariate values measured at `Time`.

```
nonEventDF2 <- dat$nonEventDF2
head(nonEventDF2)
```

```
##   ID   Time Z1ti       Z2ti       Z3ti
## 1  1 0.0000    0 -0.3792345 0.06562064
## 2  1 1.1870    1  0.8285129 0.06562064
## 3  1 2.1415    1 -0.7402917 0.06562064
## 4  2 0.0000    0  0.9967859 0.21085317
## 5  2 0.1810    0  0.8830705 0.21085317
## 6  2 0.8635    0 -0.8732199 0.21085317
```

3. **eventDF**: A data frame that record covariates for the event model **at event visits**. The first two columns must be ID and Time:

- `ID`: The subject ID for each observation;
- `Time`: Time zero and the event times for all the subjects;
- The other columns are covariate values measured at `Time`.

```
eventDF <- dat$eventDF
head(eventDF)
```

```
##   ID Time Z1ui       Z2ui        Z3ui
## 1  1    0    0 -0.3792345  0.06562064
## 2  2    0    0  0.9967859  0.21085317
## 3  3    0    1  0.1043174  0.40099376
## 4  4    0    1  0.8214735  0.09008029
## 5  5    0    0  0.3275273 -0.18271382
## 6  6    0    1  0.9684439 -0.16643427
```

Model fittng

After the data frames have been prepared, one can apply the `reVAR()` function to obtain the coefficients in the event and non-event models. To obtain the 95% confidence intervals, the function `reVARBoot()` can be applied to produce bootstrapping confidence intervals. The function returns the estimations on `B` bootstrapped datasets.

Using `reVAR()`

```
fit <- reVAR(nonEventDF1, nonEventDF2, eventDF,
             tau = 4.5, h = 0.3, baseline = TRUE)

bt <- reVARBoot(nonEventDF1, nonEventDF2, eventDF,
                tau = 4.5, h = 0.3, baseline = TRUE, B = 500)

se.beta <- apply(bt$beta, 2, sd)
se.alpha <- apply(bt$alpha, 2, sd)

# The event model
```

```r
kable(data.frame(Variable = c("Z1","Z2","Z3"),
                 Coefficient = fit$beta,
                 Lower = fit$beta - 1.96*se.beta,
                 Upper = fit$beta + 1.96*se.beta))
```

Variable	Coefficient	Lower	Upper
Z1	-0.7977177	-1.2981765	-0.2972589
Z2	0.9460350	0.5785989	1.3134711
Z3	-1.2186478	-2.0496439	-0.3876517

```r
# The non-event visit model
kable(data.frame(Variable = c("X1","X2","X3"),
                 Coefficient = fit$alpha,
                 Lower = fit$alpha - 1.96*se.alpha,
                 Upper = fit$alpha + 1.96*se.alpha))
```

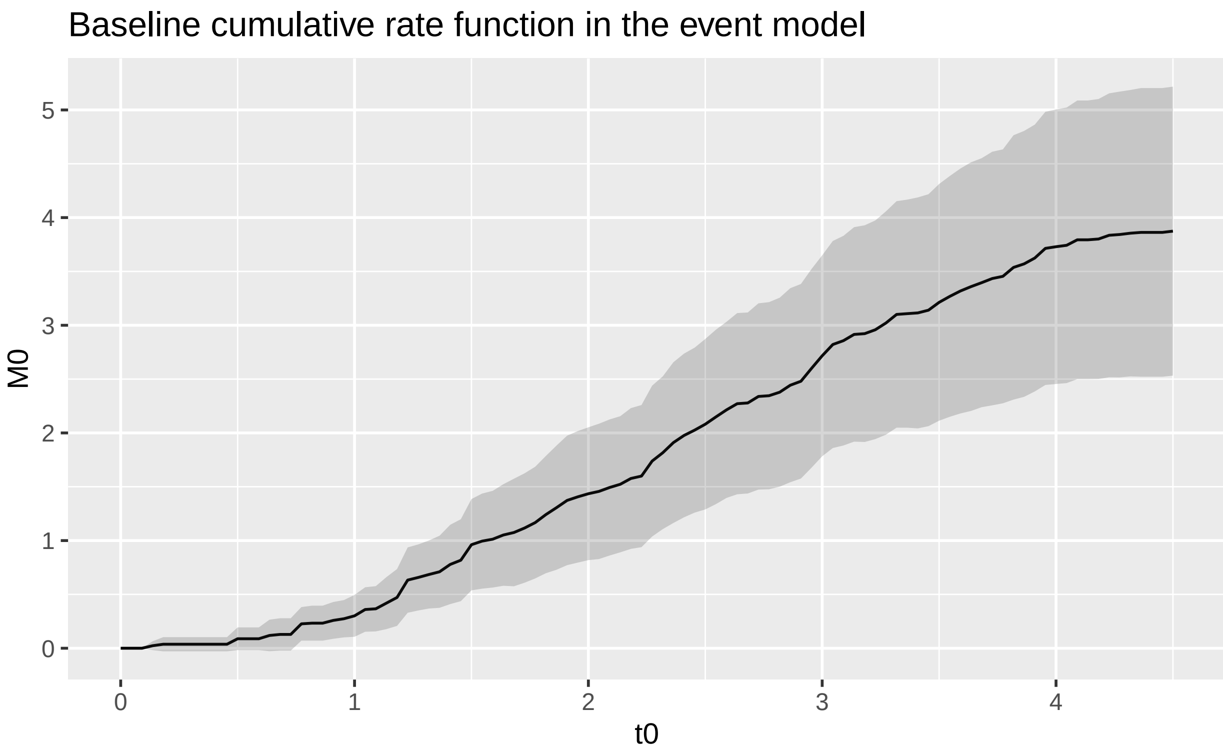
Variable	Coefficient	Lower	Upper
X1	-1.1084895	-1.3257573	-0.8912217
X2	0.9877510	0.8222388	1.1532631
X3	-0.9784618	-1.2392155	-0.7177082

Finally, one can also obtain the point estimates and pointwise 95% confidence intervals of baseline rate functions of the event and non-event visit models when setting `base = TRUE` in the functions `reVAR()` and `reVARBoot()`.

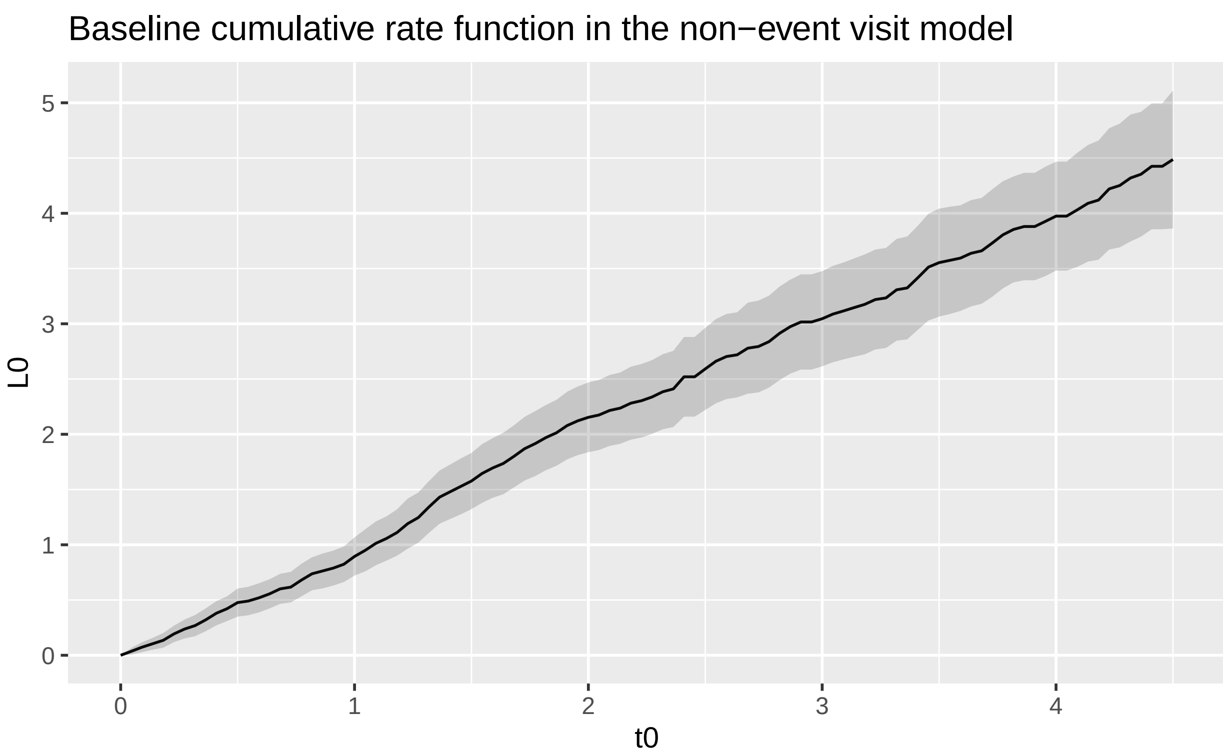
```r
se.M0 <- apply(bt$M0, 2, sd)
se.L0 <- apply(bt$L0, 2, sd)

pdata <- data.frame(t0 = fit$t0,
                    M0 = fit$M0, L0 = fit$L0,
                    M0_L = fit$M0 - 1.96*se.M0,
                    M0_U = fit$M0 + 1.96*se.M0,
                    L0_L = fit$L0 - 1.96*se.L0,
                    L0_U = fit$L0 + 1.96*se.L0)


ggplot(pdata, aes(x = t0, y = M0)) +  geom_line() +
  geom_ribbon(aes(ymin = M0_L, ymax = M0_U), alpha=0.2) +
  ggtitle("Baseline cumulative rate function in the event model")
```




```r
ggplot(pdata, aes(x = t0, y = L0)) +  geom_line() +
  geom_ribbon(aes(ymin = L0_L, ymax = L0_U), alpha=0.2) +
  ggtitle("Baseline cumulative rate function in the non-event visit model")
```



Other methods

To obtain the coefficient estimates in the event model using the methods in Li et al. (2016), we can apply the function `reVCAR()`. Only covariates in the event visit model are needed.

```
beta.VCAR <- reVCAR(nonEventDF2, eventDF, tau = 4.5, h = 0.3)

bt2 <- reVCARBoot(nonEventDF2, eventDF,  tau = 4.5, h = 0.3, B = 500)

se2.beta <- apply(bt2, 2, sd)

kable(data.frame(Variable = c("Z1","Z2","Z3"),
                 Coefficient = beta.VCAR,
                 Lower = beta.VCAR - 1.96*se2.beta,
                 Upper = beta.VCAR + 1.96*se2.beta))
```

Variable	Coefficient	Lower	Upper
Z1	-0.3381703	-0.7686943	0.0923538
Z2	0.6715065	0.3833191	0.9596938
Z3	-0.0675082	-0.7285248	0.5935083

To obtain the coefficient estimates in the event model using the last-covariate-carrying-forward (LCCF) approach, the function `coxph()` can be used. In this example, the data frame `dat$nonEventDF1_s2` contains all the information needed for the LCCF approach.

```
fit3 <- coxph(Surv(Start, End, Status2) ~ X1 + X2 + X3 + cluster(ID),
              data = dat$nonEventDF1_s2)

coef3<- summary(fit3)$coefficients

kable(data.frame(Variable = c("Z1","Z2","Z3"),
                 Coefficient = coef3[,1],
                 Lower = coef3[,1] - 1.96*coef3[,4],
                 Upper = coef3[,1] + 1.96*coef3[,4]),
      row.names = FALSE)
```

Variable	Coefficient	Lower	Upper
Z1	-0.3690577	-0.6188240	-0.1192914
Z2	0.3543437	0.1787101	0.5299774
Z3	-1.3556624	-1.7813640	-0.9299609

References

Li, S., Sun, Y., Huang, C.-Y., Follmann, D.A., & Krause, R. (2016). Recurrent event data analysis with intermittently observed time-varying covariates. Statistics in Medicine, 35(18), 3049-3065.

Sun, Y., McCulloch, C.E., Marr, K.A., Huang, C.-Y.. (2020+). Recurrent Events Analysis With Data Collected at Informative Clinical Visits in Electronic Health Records.