Relational Event Models 2.0

Reciprocity and triadic effects revisited - Tutorial Section

R. Juozaitienė, F. Bianchi, M. Boschi, E. Filippi-Mazzola, E. C. Wit

2024-06-24

Simulated Data Example

Load the Data:

```
load(file = "00-Data/simdat.RData")
head(simdat)
```

```
##   sender receiver        time           id
## 1      7       15 0.002617691 1.836660e+09
## 2      7       15 0.003056666 1.836660e+09
## 3      5        7 0.006492790 6.998400e+04
## 4     14       19 0.008528536 1.904249e+13
## 5     14       15 0.009936358 2.350925e+11
## 6      3        6 0.011259221 5.832000e+03
```

In this case, we analyze a set of interactions observed among 20 individuals. We assume that the same interaction can be repeated multiple times and that all pairs of individuals can interact, with the exception of self-loops.

Some preliminaries:

```
#Libraries that we are going to use
library(tidyr)
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
set.seed(1234)
```

To implement the maximum sampled partial likelihood, we use case-control datasets. These include a dataset of observed events (sender, receiver, timestamp) with their corresponding covariates, and a dataset of non-events, which also consist of a sender, receiver, timestamp, and corresponding covariates.

Full Risk Set

To create a set of non-events, we need to define the risk set. This involves generating a set of all possible events that can occur among 20 individuals, excluding self-loops.

```
#Number of individuals
p <- 20
risk_set <- cbind(rep(1:p,each=p),rep(1:p,p))
#Remove cases where sender = receiver
t <- which(risk_set[,1] == risk_set[,2])
```

```
risk_set <- risk_set[-t,]
risk_set <- as.data.frame(risk_set)
colnames(risk_set) <- c("sender","receiver")

#Add unique ID to each pair
risk_set$id <- 2^risk_set$sender*3^risk_set$receiver
head(risk_set)
```

```
##   sender receiver   id
## 1      1        2   18
## 2      1        3   54
## 3      1        4  162
## 4      1        5  486
## 5      1        6 1458
## 6      1        7 4374
```

Sampled Risk Set

For each observed event, we need to sample one non-event. This means that from the full risk set, we sample one event that does not match the observed event at that time.

```
non.id <- NULL
#For each event sample a non-event, i.e. its ID
for(i in 1:nrow(simdat))
  non.id <- c(non.id,sample(risk_set$id[risk_set$id!=simdat$id[i]],1))
```

Vector `non.id` contains the IDs of the non-events. This information enables us to extract the sender and receiver IDs. To facilitate this process, we create an additional vector containing row numbers corresponding to these non-events.

```
row_n <- NULL
for(i in 1:length(non.id))
{
  row_n <- c(row_n,which(risk_set$id == non.id[i]))
}
```

Finally, we create a dataset of non-events containing sender, receiver and timestamp information. The timestamps for the non-events coincide with those of the events.

```
#Timestamps for the non-events
non.tm <- simdat$time
#Create a dataset of non-events consisting of sender,receiver and time
non.data <- as.data.frame(cbind(risk_set$sender[row_n],risk_set$receiver[row_n],non.tm))
colnames(non.data)[1:3] <- c("sender","receiver","time")
```

Endogenous Network Effects

To compute transitivity statistics based on different definitions, we utilize a function called `n_effects`. Its input requires two dataset corresponding to the sets of events and non-events, respectively. In each dataset, the first column denotes the sender, the second column the receiver, and the third column represents the timestamp. The function also includes a third argument `T` which is a half-life parameter in an exponential decaying function determining the rate at which the weights of past events are reduced. If this value is not given, the default value is defined as the average waiting time between events, calculated as:

$$\frac{(\max(t) - \min(t))n}{p(p-1)},$$

where t is the time vector, p is the number of individuals and n is number of events.

```
rez <- n_effects(simdat,non.data)
simdat <- rez[[1]]
non.data <- rez[[2]]
```

If the estimation takes too much time, you can directly upload the full data set via:

```
load("00-Data/solution.RData")
```

The output of this function is a list containing the initial datasets augmented with the statistics of reciprocity and transitivity effects.

Reciprocity:

- r1 - binary effect

$$r_{sr}^{(1)}(t) = \begin{cases} 1 & \text{if } (r,s) \text{ happened before,} \\ 0 & \text{otherwise.} \end{cases}$$

- r2 - effect accounting for the volume of past events

$$r_{sr}^{(2)}(t) = \sum_{t_i<t} 1_{\{s_i=r,r_i=s\}}.$$

- r3 - effect defined via exponential decaying function

$$r_{sr}^{(3)}(t) = \sum_{i:s_i=r,r_i=s,t_i<t} \mathrm{e}^{-(t-t_i)\frac{\ln 2}{T}} \frac{\ln 2}{T}.$$

- r4a - data-driven definition, assuming that the time to reciprocation is defined as the difference between the time of the current event t and the most recent event in the opposite direction (r,s),

$$r_{sr}^{(4a)}(t) = f^R(\Delta t_{sr}^{\mathrm{rec}(a)}),$$

  and $\Delta t_{sr}^{\mathrm{rec}(a)} = \begin{cases} t - t_e & \text{where } t_e = \max\{t_i < t \mid s_i = r, r_i = s\} \\ \infty & \text{in case } \{t_i < t \mid s_i = r, r_i = s\} = \emptyset. \end{cases}$

- r4b - data-driven definition, assuming that the time to reciprocation is defined as the difference between the time of the current event and the first event (r,s),

$$r_{sr}^{(4b)}(t) = f^R(\Delta t_{sr}^{\mathrm{rec}(b)}),$$

  and $\Delta t_{sr}^{\mathrm{rec}(b)} = \begin{cases} t - t_e & \text{where } t_e = \min\{t_i < t \mid s_i = r, r_i = s\} \\ \infty & \text{in case, } \{t_i < t \mid s_i = r, r_i = s\} = \emptyset. \end{cases}$

Transitivity:

- t1 - binary effect

$$t_{sr}^{(1)}(t) = \begin{cases} 1 & \exists k \in V : (s,k) \text{ and } (k,r) \text{ happened before } t \\ 0 & \text{otherwise.} \end{cases}$$

- t2 - effect accounting for the volume of past events

$$t_{sr}^{(2)}(t) = \sum_{k \in V} 1_{\{t_i<t,s_i=s,r_i=k\} \wedge \{t_j<t,s_j=k,r_j=r\}}.$$

- t3 - effect defined via exponential decaying function

$$t_{sr}^{(3)}(t) = \sum_{\substack{k \in V: \\ (s,k,t_i),t_i<t \\ (k,r,t_j),t_j<t}} \mathrm{e}^{-\left(t-\max(t_i,t_j)\right)\frac{\ln 2}{T}} \frac{\ln 2}{T}.$$

- t4a - data-driven definition, assuming that the time to transitive closure can be defined as the time elapsed from the most recently opened two-path:

$$t_{sr}^{(4a)}(t) = f^T(\Delta t_{sr}^{\text{tri}(a)}),$$

and $\Delta t_{sr}^{\text{tri}(a)} = \begin{cases} t - t_i & \text{where } t_i \text{ time most recent two-path } s \longrightarrow k \longrightarrow r \\ \infty & \text{otherwise.} \end{cases}$

- t4b - data-driven definition, assuming that the time to transitive closure can be defined as the time elapsed from the creation of the first two-path between s and r

$$t_{sr}^{(4b)}(t) = f^T(\Delta t_{sr}^{\text{tri}(b)}),$$

and $\Delta t_{sr}^{\text{tri}(b)} = \begin{cases} t - t_i & \text{where } t_i \text{ time first two-path } s \longrightarrow k \longrightarrow r \\ \infty & \text{otherwise.} \end{cases}$

These definitions can also be adjusted to take into account ordering of the events. We will refer to them as t1o, t2o, t3o, t4ao, t4bo, respectively.

Modelling

Create a target variable consisting of values equal to 1.

```
#Target variable
y<-rep(1,nrow(simdat))
```

Estimate transitivity using binary effect in case when order is not important:

```
#Define a variable as the difference between event and non-event
x.t1 <- simdat$t1-non.data$t1
#Fit the model
gam.t1<-gam(y~-1+ x.t1,family=binomial)
#Model summary
summary(gam.t1)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ -1 + x.t1
##
## Parametric coefficients:
##      Estimate Std. Error z value Pr(>|z|)
## x.t1 -0.29775    0.04488  -6.634 3.27e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =   -Inf   Deviance explained = -Inf%
## UBRE = 0.38204  Scale est. = 1         n = 10000
```

Estimate transitivity using effect accounting for number of third parties involved and assuming that order of the events is not important:

```
x.t2 <- simdat$t2-non.data$t2
gam.t2<-gam(y~-1+ x.t2,family=binomial)
#Model summary
summary(gam.t2)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ -1 + x.t2
##
## Parametric coefficients:
##      Estimate Std. Error z value Pr(>|z|)
## x.t2 0.065993   0.002603   25.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =   -Inf   Deviance explained = -Inf%
## UBRE = 0.31663  Scale est. = 1         n = 10000
```

Estimate transitivity using effect defined via exponential decaying function

```
x.t3  <- simdat$t3-non.data$t3
gam.t3<-gam(y~-1+ x.t3,family=binomial)
#Model summary
summary(gam.t3)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ -1 + x.t3
##
## Parametric coefficients:
##      Estimate Std. Error z value Pr(>|z|)
## x.t3   54.614      2.155   25.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =   -Inf   Deviance explained = -Inf%
## UBRE = 0.31667  Scale est. = 1         n = 10000
```

Estimate transitivity using data-driven definition, assuming that the time to transitive closure can be defined as the time elapsed from the creation of the first two-path and disregarding the order of the occurrence of links:

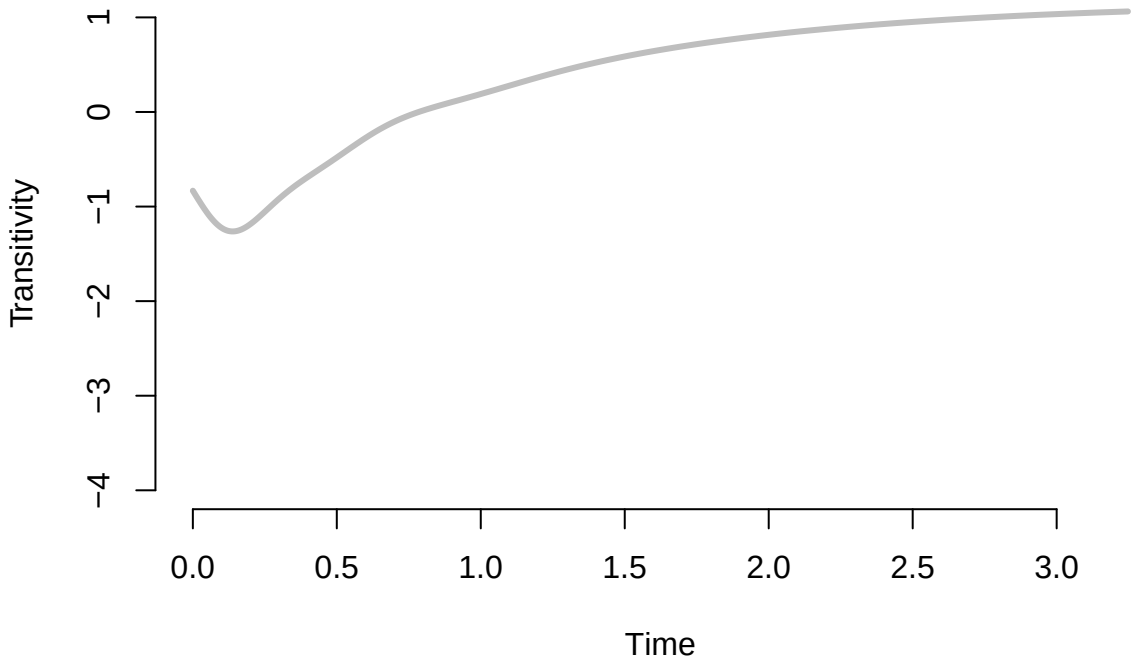
```
#Smooth function of time defined as the most distant triad
#Construct the object combining statistics values for events and non-events.
tm.mat<-cbind(simdat$t4b,non.data$t4b)
#Construct the object identifying whether events and non-events were transitive ones:
id.mat<-cbind(simdat$t1,-non.data$t1)
#Model fitting
gam.t4b<-gam(y~ -1+ s(tm.mat, by=id.mat) ,family = binomial)
#Model summary
summary(gam.t4b)
```

```
##
```

```
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ -1 + s(tm.mat, by = id.mat)
##
## Approximate significance of smooth terms:
##                     edf Ref.df Chi.sq p-value
## s(tm.mat):id.mat 7.967  9.055   1214  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   -Inf   Deviance explained = -Inf%
## UBRE = 0.24164  Scale est. = 1         n = 10000
```

The estimated effect is on the transformed time scale. To return it to the original scale, we need additional steps.

```
#Create a sequence of equally spaced time points
ts<-seq(min(simdat$t4b[simdat$t4b!=0]),max(simdat$t4b),length=200)
#Create an object based on these generated time points
ndat<-data.frame(tm.mat=ts,id.mat=rep(1,200))
#Use gam model to get estimates for these time points
pred<-predict(gam.t4b,newdata = ndat,type="terms")
#Plot the results taking into account that the time was transformed via exp(-t)
plot(-log(ts),pred[,1],type="l", col='grey', lwd = 3, bty='n',
     ylim = c(-4,1), ylab = "Transitivity", xlab ="Time")
```



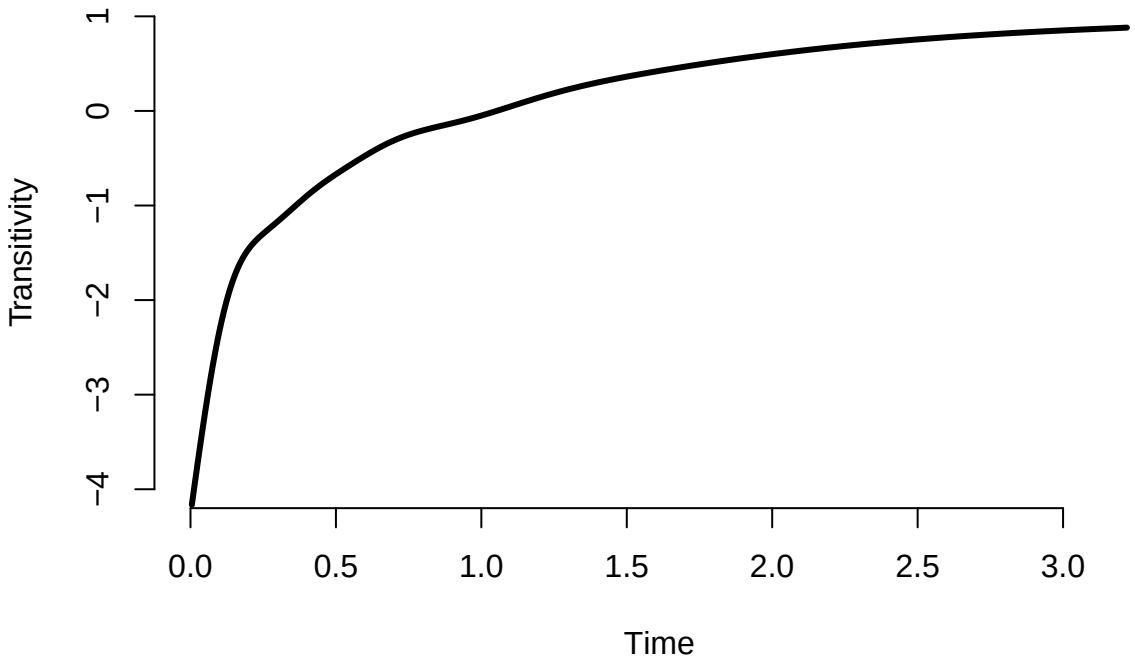
Estimate transitivity using the same data-driven definition, but assuming that the order of the occurrence of links is important:

```
#Smooth function of time defined as the most distant triad
tm.mat2<-cbind(simdat$t4bo,non.data$t4bo)
id.mat2<-cbind(simdat$t1o,-non.data$t1o)

#Model fitting
gam.t4bo<-gam(y~ -1+ s(tm.mat2, by=id.mat2) ,family = binomial)

#Model summary
summary(gam.t4bo)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ -1 + s(tm.mat2, by = id.mat2)
##
## Approximate significance of smooth terms:
##                       edf Ref.df Chi.sq p-value
## s(tm.mat2):id.mat2 9.254  9.846   1499  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   -Inf   Deviance explained = -Inf%
## UBRE = 0.14693  Scale est. = 1         n = 10000
```

Additional steps for plotting the estimate:

```
ts2<-seq(min(simdat$t4bo[simdat$t4bo!=0]),max(simdat$t4bo),length=200)
ndat2<-data.frame(tm.mat2=ts2,id.mat2=rep(1,200))
pred2<-predict(gam.t4bo,newdata = ndat2,type="terms")
#Plot the estimate of transitivity
plot(-log(ts2),pred2[,1],type="l", lwd = 3, bty='n',
     ylim = c(-4,1), ylab = "Transitivity", xlab ="Time")
```



AIC Comparison

Compare model using AIC:

```
AIC(gam.t1)
```

```
## [1] 13820.45
```

```
AIC(gam.t2)
```

```
## [1] 13166.28
```

```
AIC(gam.t3)
```

```
## [1] 13166.7
```

```
AIC(gam.t4b)
```

```
## [1] 12416.43
```

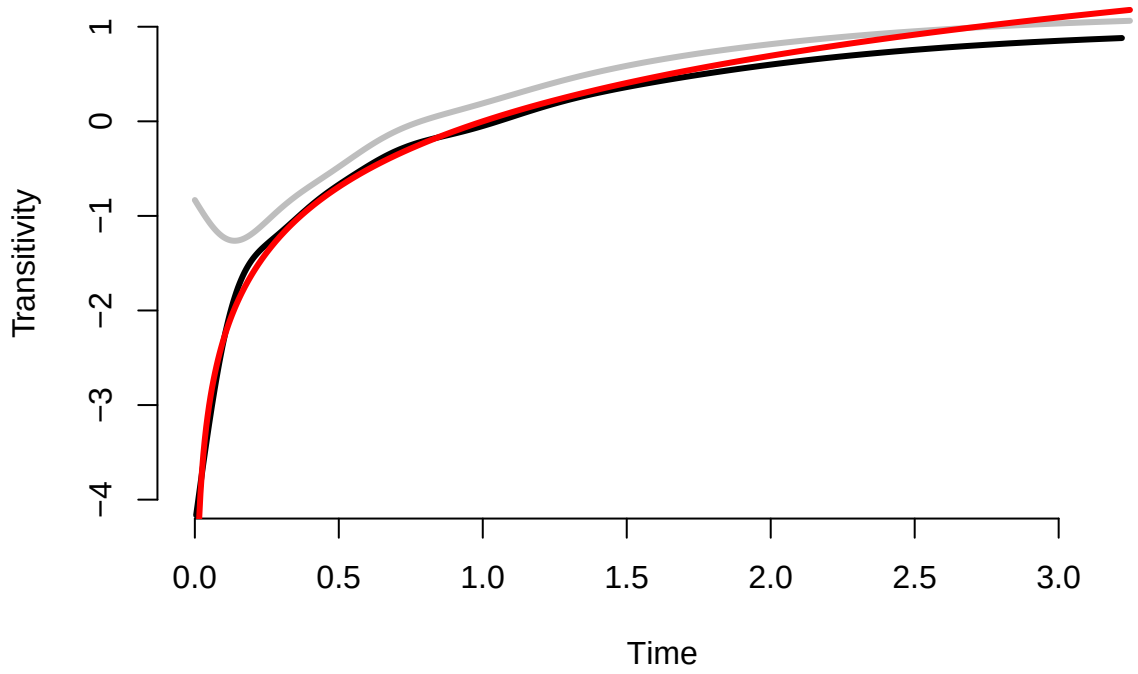
```
AIC(gam.t4bo)
```

```
## [1] 11469.33
```

The Actual Effect of Transitivity:

```
plot(-log(ts),pred[,1],type="l", col='grey', lwd = 3, bty='n',
     ylim = c(-4,1), ylab = "Transitivity", xlab ="Time")
```

```
lines(-log(ts2),pred2[,1],type="l",  lwd = 3)
lines(-log(ts),log(-log(ts)),col="red", lwd = 3)
```



Exercise: Classroom data

Load the input data, it is the same classroom data as you used before during endogenous covariates tutorial. We employ the same case-control datasets contained in objects *events_alt* and *no_events_alt*:

```
load(file = "00-Data/inputClass.RData")
head(events_alt)
```

```
##   sender receiver  time r1 r2         r3       r4a       r4b t1 t2 t3 t4a t4b
## 1     11        2 0.143  0  0 0.00000000 0.0000000 0.0000000  0  0  0   0   0
## 2      2       11 0.286  1  1 0.01769373 0.8667541 0.8667541  0  0  0   0   0
## 3      2        5 0.429  0  0 0.00000000 0.0000000 0.0000000  0  0  0   0   0
## 4      5        2 0.571  1  1 0.01769404 0.8676213 0.8676213  0  0  0   0   0
## 5      9        8 0.714  0  0 0.00000000 0.0000000 0.0000000  0  0  0   0   0
## 6      8        9 0.857  1  1 0.01769373 0.8667541 0.8667541  0  0  0   0   0
##   t1o t2o t3o t4ao t4bo
## 1   0   0   0    0    0
## 2   0   0   0    0    0
## 3   0   0   0    0    0
## 4   0   0   0    0    0
## 5   0   0   0    0    0
## 6   0   0   0    0    0
```

```
head(non_events_alt)
```

```
##   sender receiver non.tm r1 r2 r3 r4a r4b t1 t2 t3 t4a t4b t1o t2o t3o t4ao
## 1     15       10  0.143  0  0  0   0   0  0  0  0   0   0   0   0   0    0
## 2     11        8  0.286  0  0  0   0   0  0  0  0   0   0   0   0   0    0
## 3     10       18  0.429  0  0  0   0   0  0  0  0   0   0   0   0   0    0
## 4     15       16  0.571  0  0  0   0   0  0  0  0   0   0   0   0   0    0
## 5     18       11  0.714  0  0  0   0   0  0  0  0   0   0   0   0   0    0
## 6     18        5  0.857  0  0  0   0   0  0  0  0   0   0   0   0   0    0
##   t4bo
## 1    0
## 2    0
## 3    0
## 4    0
## 5    0
## 6    0
```

Exercise 1: Estimate the effect of reciprocity using definition *r4a*.

Exercise 2: Enrich the model by including transitivity using definition *t4a*.

Nodal Heterogeneity

We analyze a set of relational events in order to investigate the effect of transitivity on their occurrence. The provided data includes fully prepared datasets for both events (*simdat*) and non-events (*non.data*) each encompassing all relevant endogenous network effects of interest. Load the Data:

```
load(file = "00-Data/re_data.RData")
head(simdat)
```

```
##   sender receiver         time           id r1 r2 r3 r4a r4b t1 t2 t3 t4a t4b
## 1      7        1 5.132239e-05 3.840000e+02  0  0  0   0   0  0  0  0   0   0
## 2     14       11 6.762887e-04 2.902376e+09  0  0  0   0   0  0  0  0   0   0
## 3      7       15 2.020945e-03 1.836660e+09  0  0  0   0   0  0  0  0   0   0
## 4      7       10 4.183139e-03 7.558272e+06  0  0  0   0   0  0  0  0   0   0
## 5     14       17 4.491085e-03 2.115832e+12  0  0  0   0   0  0  0  0   0   0
## 6     15       17 4.894437e-03 4.231665e+12  0  0  0   0   0  0  0  0   0   0
##   t1o t2o t3o t4ao t4bo       s_out        s_in r_out        r_in rep
## 1   0   0   0    0    0 0.000000000 0.000000000     0 0.000000000   0
## 2   0   0   0    0    0 0.000000000 0.000000000     0 0.000000000   0
## 3   0   0   0    0    0 0.002231893 0.000000000     0 0.000000000   0
## 4   0   0   0    0    0 0.004463775 0.000000000     0 0.000000000   0
## 5   0   0   0    0    0 0.002231884 0.000000000     0 0.000000000   0
## 6   0   0   0    0    0 0.000000000 0.002231889     0 0.002231901   0
##    turntaking   turnconti
## 1 0.000000000 0.000000000
## 2 0.000000000 0.000000000
## 3 0.000000000 0.002231893
## 4 0.000000000 0.002231892
## 5 0.000000000 0.002231884
## 6 0.002231889 0.000000000
```

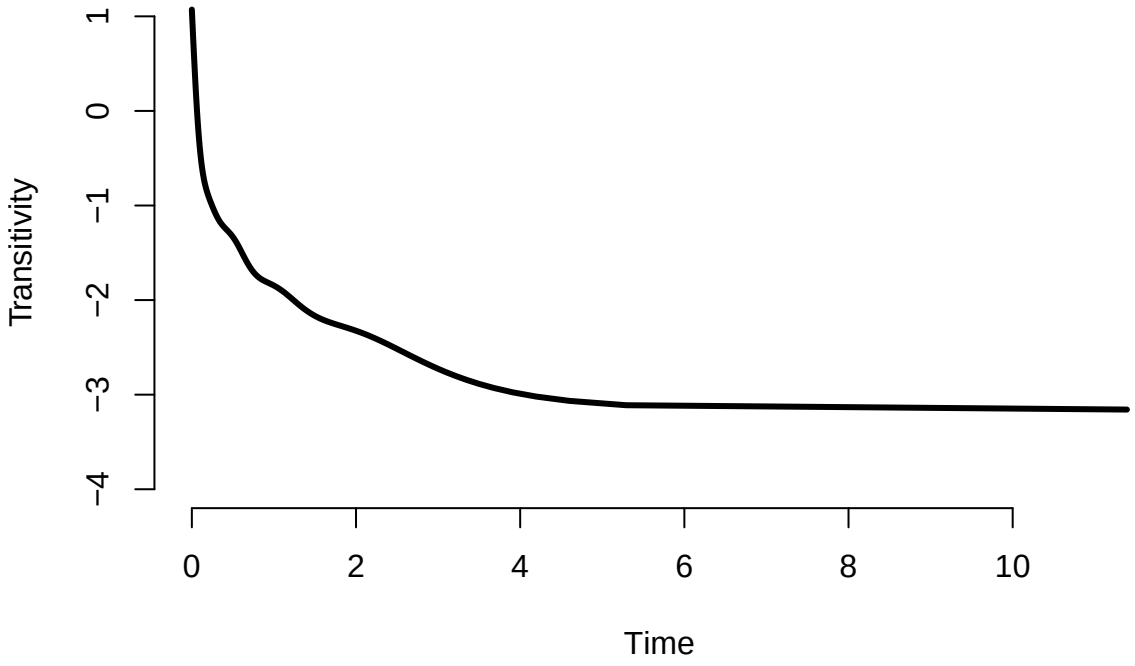
Based on previous examples we will focus on the data-driven definition of transitivity. Thus we will fit a model using definiton *t4b* which assumes that the time to transitive closure can be defined as the time elapsed from the creation of the first two-path:

```r
#Target variable
y<-rep(1,nrow(simdat))

tm.mat<-cbind(simdat$t4b,non.data$t4b)
id.mat<-cbind(simdat$t1,-non.data$t1)

#Model fitting
fit <-gam(y~ -1+ s(tm.mat, by=id.mat) ,family = binomial)
#Model summary
summary(fit)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ -1 + s(tm.mat, by = id.mat)
##
## Approximate significance of smooth terms:
##                     edf Ref.df Chi.sq p-value
## s(tm.mat):id.mat 9.149    9.8   2461  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   -Inf   Deviance explained = -Inf%
## UBRE = -0.049266  Scale est. = 1         n = 10000
```

Plot the estimate of transitivity:

```r
#To get a smooth estimate over the time interval of interest
#Create a sequence of equally spaced time points
ts<-seq(min(simdat$t4b[simdat$t4b!=0]),max(simdat$t4b),length=200)
#Create an object based on these generated time points
ndat<-data.frame(tm.mat=ts,id.mat=rep(1,200))
#Using gam model get estimates for these specific time points
pred<-predict(fit,newdata = ndat,type="terms")
#Plot the results taking into account that the time was transformed via exp(-t)
plot(-log(ts),pred[,1],type="l", lwd = 3, bty='n', ylim = c(-4,1),
     ylab = "Transitivity", xlab ="Time")
```



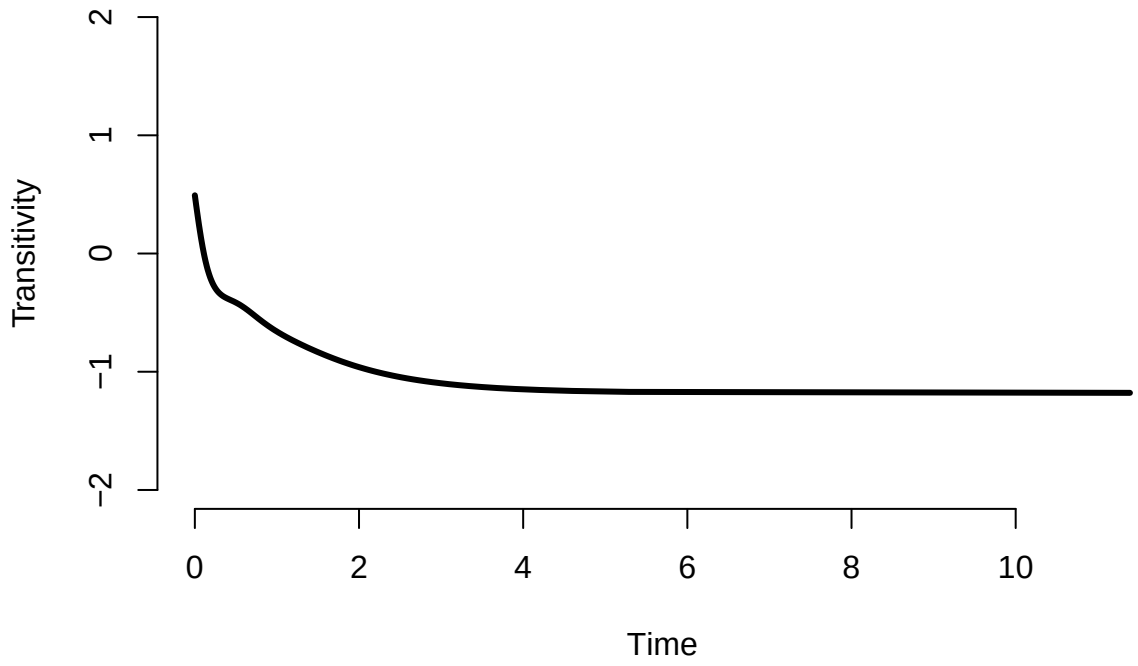
This model violates the Hierarchy principle. Therefore, we will enrich the model by including 1st-order effects, specifically sender and receiver in-degree and out-degree.

```
#Define model covariates
s_out <- simdat$s_out - non.data$s_out
s_in <- simdat$s_in - non.data$s_in
r_out <- simdat$r_out - non.data$r_out
r_in <- simdat$r_in - non.data$r_in

#Fit the model
fit2 <-gam(y~-1+ s_out + s_in + r_out + r_in + s(tm.mat, by=id.mat) ,family = binomial)
summary(fit2)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ -1 + s_out + s_in + r_out + r_in + s(tm.mat, by = id.mat)
##
## Parametric coefficients:
##        Estimate Std. Error z value Pr(>|z|)
## s_out  0.849186   0.037291  22.772  < 2e-16 ***
## s_in  -0.197953   0.041362  -4.786  1.7e-06 ***
## r_out -0.005995   0.040200  -0.149    0.881
## r_in   0.466324   0.017527  26.606  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                     edf Ref.df Chi.sq p-value
## s(tm.mat):id.mat 5.858  6.922  286.3  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   -Inf   Deviance explained = -Inf%
## UBRE = -0.25028  Scale est. = 1         n = 10000
```

```
ndat2<-data.frame(tm.mat=ts,id.mat=rep(1,200),
                  s_out = rep(1,200),r_out = rep(1,200),
                  s_in = rep(1,200), r_in = rep(1,200))
pred2<-predict(fit2,newdata = ndat2,type="terms")
plot(-log(ts),pred2[,5],type="l", lwd = 3, bty='n', ylim = c(-2,2),
     ylab = "Transitivity", xlab ="Time")
```



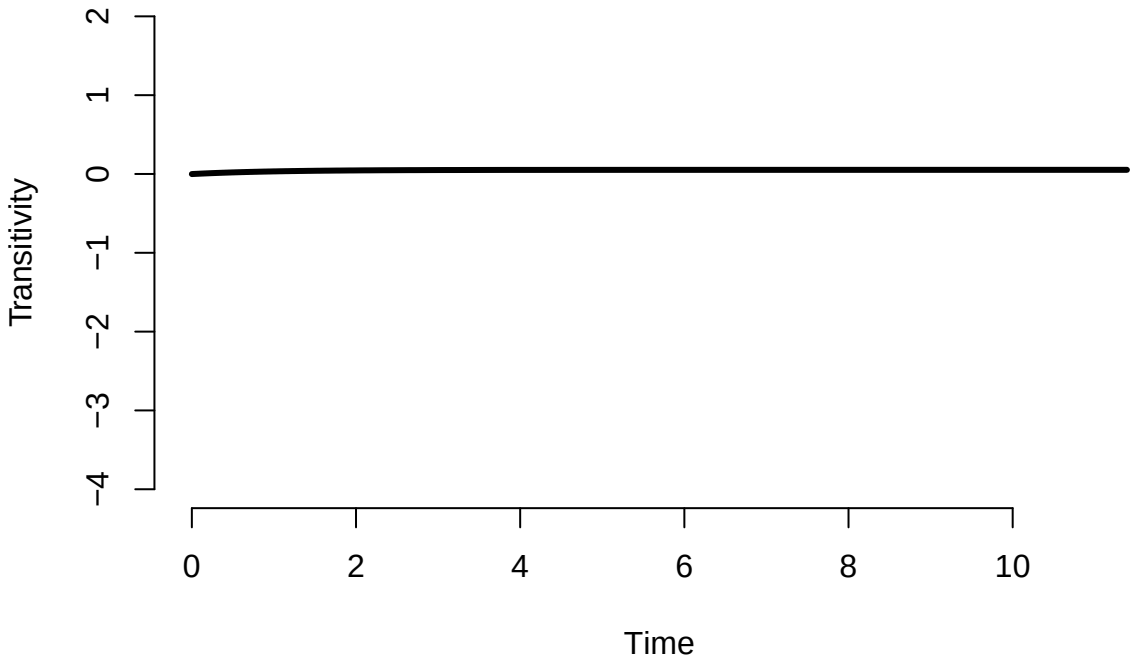
We will compare this estimate with the one obtained via model including both transitivity and random nodal effects:

```
sender_re <- factor(cbind(simdat$sender,non.data$sender))
dim(sender_re) <- c(nrow(simdat),2)

receiver_re <- factor(cbind(simdat$receiver,non.data$receiver))
dim(receiver_re) <- c(nrow(simdat),2)
```

```r
id <-cbind(rep(1,nrow(simdat)),rep(-1,nrow(simdat)))

fit3<-gam(y~-1+ + s_out + s_in + r_out + r_in
              + s(tm.mat, by=id.mat)
              + s(sender_re, by = id, bs = 're')
              + s(receiver_re, by = id, bs = 're'),family=binomial)
summary(fit3)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## y ~ -1 + +s_out + s_in + r_out + r_in + s(tm.mat, by = id.mat) +
##     s(sender_re, by = id, bs = "re") + s(receiver_re, by = id,
##     bs = "re")
##
## Parametric coefficients:
##        Estimate Std. Error z value Pr(>|z|)
## s_out -0.027949   0.061358  -0.456    0.649
## s_in  -0.097269   0.072882  -1.335    0.182
## r_out  0.103327   0.079507   1.300    0.194
## r_in   0.008043   0.026940   0.299    0.765
##
## Approximate significance of smooth terms:
##                       edf Ref.df   Chi.sq p-value
## s(tm.mat):id.mat     2.00  2.001    0.311   0.856
## s(sender_re):id     18.84 19.000 1180.353  <2e-16 ***
## s(receiver_re):id 18.85 19.000 1679.420  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   -Inf   Deviance explained = -Inf%
## UBRE = -0.40151  Scale est. = 1         n = 10000
```

```r
ndat3<-data.frame(tm.mat=ts,id.mat=rep(1,200),
                  s_out = rep(1,200),r_out = rep(1,200),
                  s_in = rep(1,200), r_in = rep(1,200),
                  sender_re = rep(1,200),id = rep(1,200),
                  receiver_re = rep(1,200))
pred3<-predict(fit3,newdata = ndat3,type="terms")
plot(-log(ts),pred3[,5],type="l", lwd = 3, bty='n', ylim = c(-4,2),
     ylab = "Transitivity", xlab ="Time")
```


Transitivity
2
1
0
−1
−2
−3
−4
0
2
4
6
8
10
Time