Saemix 3 - count data models

Emmanuelle

08/2022

Version

Use saemix version ≥ 3.2

Objective

Run binary and categorical models in **saemix**

This notebook uses additional code from the **saemix** development github, not yet integrated in the package. The *workDir* folder in the next chunk of code points to the folder where the user stored this code, and is needed to run the notebook (*workDir* defaults to the current working directory). Specifically, the notebook loads:

- code for different bootstraps in non-linear mixed effect models (Comets et al. 2021 and submitted)
  - the bootstrap runs have been performed previously and are stored in files to be read
    - bootstraps can be run instead by switching the *runBootstrap* variable to TRUE in the first chunk of code
    - in the code, the number of bootstraps is set to 10 for speed but we recommend to use at least 200 for a 90% CI.
  - this can be changed in the following change of code by uncommenting the line *nboot<-200* and setting the number of bootstrap samples (this may cause memory issues in **Rstudio** with older machines, if this is the case we recommend executing the code in a separate script)
- code for the MC/AGQ provided by Sebastian Ueckert (Ueckert et al. 2017)
  - again if memory issues arise the code can be run in a separate script.

The current notebook can be executed to create an HMTL or PDF output with comments and explanations. A script version containing only the R code is also given as *saemix3_categoricalModel.R* in the same folder.

Count data model

Data description

- Drinking patterns amongst students (David Atkins from tutorial)
  - dataset rapi.saemix
  - lambda parameter from Poisson model with a time-effect, gender effects on both intercept and slope
  - different models can be adjusted to the data, accounting for overdispersion **Copy from documentation, describing papers from Atkins' group**

The *rapi.saemix* dataset in the **saemix** package contains count data kindly made available by David Atkins (University of Washington) in his tutorial on modelling count data (Atkins et al. 2013). The data comes from a randomised controlled trial assessing the effectiveness of web-based personalised normative feedback intervention on alcohol consumption (Neighbors et al. 2010a, 2010b}. The *rapi.saemix* dataset records alcohol-related problems, as measured by the Rutgers Alcohol Problem Index (RAPI) (White et al. 1989), in freshmen at risk for heavy drinking behaviours. Students were asked to report every six months the number of alcohol-related problems, and the dataset includes 3,616 repeated measures of these counts in 818 subjects, 561 of whom had the full 5 measurements over a period of 2 years. Interesting features of this dataset are first, the longitudinal aspect which allow to evaluate changes over time, and second, the shape of the distribution of counts. Counts are often positively skewed, bounded by zero, with a large stack of data points at zero, indicating individuals and/or occasions without drinking, use, or related problems. This dataset was used in Atkins et al. (2013) to illustrate mixed effects count regression using the *glmer()* function from the **lme4** package.

Similarly to categorical data, we need the value of the outcome to compute the associated likelihood. Therefore, to create the data object using saemixData, we need to specify the response column both as a response (*name.response="rapi"*) and as a predictor (here, time is the first predictor and we add the response in the argument *name.predictors*).

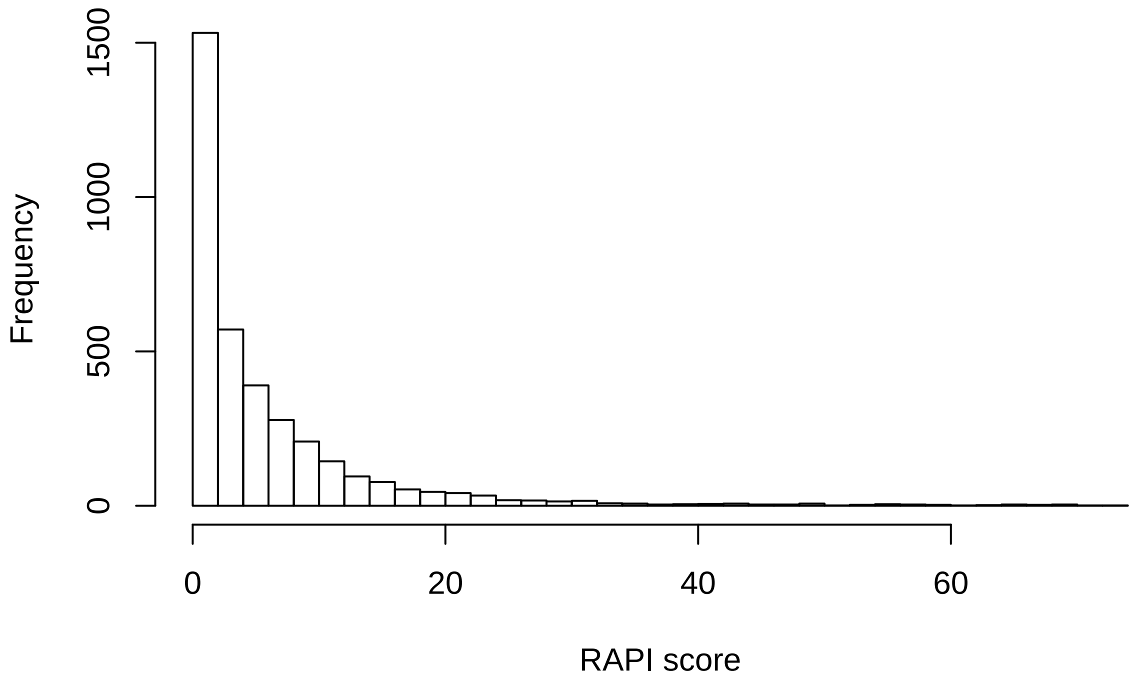
```
data(rapi.saemix)

saemix.data<-saemixData(name.data=rapi.saemix, name.group=c("id"),
                        name.predictors=c("time","rapi"),name.response=c("rapi"),
                        name.covariates=c("gender"),
                        units=list(x="months",y="",covariates=c("")))
```

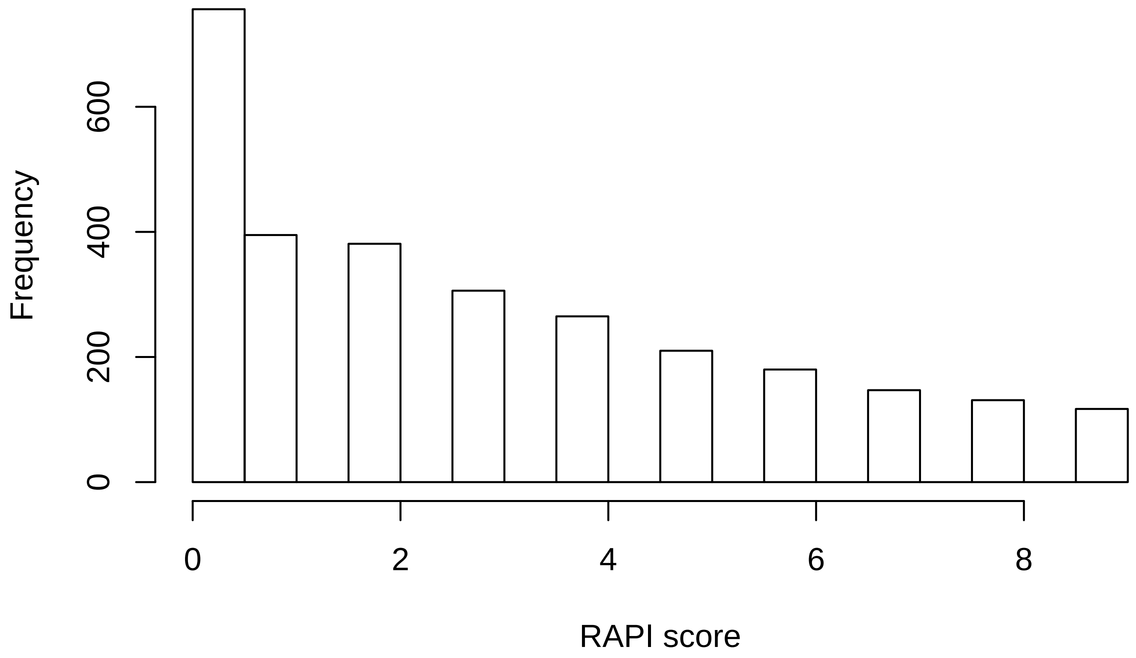
```
##
##
## The following SaemixData object was successfully created:
##
## Object of class SaemixData
##     longitudinal data for use with the SAEM algorithm
## Dataset rapi.saemix
##     Structured data: rapi ~ time + rapi | id
##     X variable for graphs: time (months)
##     covariates: gender ()
##       reference class for covariate gender :  Men
```

Exploring data The distribution of count data can be visualised as a histogram. The over-representation of low scores can be seen when zooming on the early part of the histogram. We can also tabulate the data by stratifying on men and women to realise that there seems to be a gender difference, with more women not reporting any episode of drinking than men.

```
# Simple histogram
hist(rapi.saemix$rapi, main="", xlab="RAPI score", breaks=30)
```



```
# Zooming on small values of scores
hist(rapi.saemix$rapi[rapi.saemix$rapi < 10], main="", xlab="RAPI score", breaks=30)
```



```
table(rapi.saemix$gender, as.integer(rapi.saemix$rapi > 2))
```

```
##
##            0    1
##   Men    548  938
##   Women  984 1146
```

Statistical model Several models can be fit to the data

- the simplest one is a Poisson model
    - below, we directly include a time effect on the λ parameter of the Poisson model, which is defined as a linear function of time.
- several models can

For the statistical model, we assume a normal distribution for intercept and slope, so that the distribution of λ is log-normal.

```r
## Poisson with a time effect
# Model
count.poisson<-function(psi,id,xidep) {
  time<-xidep[,1]
  y<-xidep[,2]
  intercept<-psi[id,1]
  slope<-psi[id,2]
  lambda<- exp(intercept + slope*time)
  logp <- -lambda + y*log(lambda) - log(factorial(y))
  return(logp)
}
# Simulation function
countsimulate.poisson<-function(psi, id, xidep) {
  time<-xidep[,1]
  y<-xidep[,2]
  intercept<-psi[id,1]
  slope<-psi[id,2]
  lambda<- exp(intercept + slope*time)
  y<-rpois(length(time), lambda=lambda)
  return(y)
}

## Zero-inflated Poisson model
# Model
count.poissonzip<-function(psi,id,xidep) {
  time<-xidep[,1]
  y<-xidep[,2]
  intercept<-psi[id,1]
  slope<-psi[id,2]
  p0<-psi[id,3] # Probability of zero's
  lambda<- exp(intercept + slope*time)
  logp <- log(1-p0) -lambda + y*log(lambda) - log(factorial(y)) # Poisson
  logp0 <- log(p0+(1-p0)*exp(-lambda)) # Zeroes
  logp[y==0]<-logp0[y==0]
  return(logp)
}
# Simulation function
countsimulate.poissonzip<-function(psi, id, xidep) {
  time<-xidep[,1]
  y<-xidep[,2]
  intercept<-psi[id,1]
  slope<-psi[id,2]
  p0<-psi[id,3] # Probability of zero's
  lambda<- exp(intercept + slope*time)
  prob0<-rbinom(length(time), size=1, prob=p0)
  y<-rpois(length(time), lambda=lambda)
  y[prob0==1]<-0
  return(y)
}

## Generalized Poisson model with time effect
# Model
count.genpoisson<-function(psi,id,xidep) {
  time<-xidep[,1]
  y<-xidep[,2]
  intercept<-psi[id,1]
  slope<-psi[id,2]
  lambda<- exp(intercept + slope*time)
  delta<-psi[id,3]
  logp <- log(lambda) + (y-1)*log(lambda+y*delta) - lambda - y*delta - log(factorial(y))
  return(logp)
}
# Simulation function

## Negative binomial model with time effect
# Model
count.NB<-function(psi,id,xidep) {
  time<-xidep[,1]
  y<-xidep[,2]
  intercept<-psi[id,1]
  slope<-psi[id,2]
  k<-psi[id,3]
  lambda<- exp(intercept + slope*time)
  logp <- log(factorial(y+k-1)) - log(factorial(y)) - log(factorial(k-1)) + y*log(lambda) - y*log(lambda
  return(logp)
}
# Simulation function
```

Fitting models

Poisson model The code below fits the base Poisson model as well as a Poisson model with a covariance between the two parameters and a gender effect on both parameters.

```
## Poisson
### Model without covariate
saemix.model.poi<-saemixModel(model=count.poisson,description="Count model Poisson",simulate.function=co
                              modeltype="likelihood",
                              psi0=matrix(c(log(5),0.01),ncol=2,byrow=TRUE,dimnames=list(NULL, c("inter
                              transform.par=c(0,0), omega.init=diag(c(0.5, 0.5)))
```

```
##
##
## The following SaemixModel object was successfully created:
##
## Nonlinear mixed-effects model
##   Model function:  Count model Poisson
##   Model type:  likelihood
## function(psi,id,xidep) {
##   time<-xidep[,1]
##   y<-xidep[,2]
##   intercept<-psi[id,1]
##   slope<-psi[id,2]
##   lambda<- exp(intercept + slope*time)
##   logp <- -lambda + y*log(lambda) - log(factorial(y))
##   return(logp)
## }
##   Nb of parameters: 2
##       parameter names:  intercept slope
##       distribution:
##      Parameter Distribution Estimated
## [1,] intercept normal       Estimated
## [2,] slope     normal       Estimated
##   Variance-covariance matrix:
##           intercept slope
## intercept         1     0
## slope             0     1
##     No covariate in the model.
##     Initial values
##              intercept slope
## Pop.CondInit  1.609438  0.01
```

```
### Gender effect on intercept and slope
saemix.model.poi.cov2<-saemixModel(model=count.poisson,description="Count model Poisson",simulate.functi
                                   modeltype="likelihood",
                                   psi0=matrix(c(log(5),0.01),ncol=2,byrow=TRUE,dimnames=list(NULL, c("
                                   transform.par=c(0,0), omega.init=diag(c(0.5, 0.5)),
                                   covariance.model =matrix(data=1, ncol=2, nrow=2),
                                   covariate.model=matrix(c(1,1), ncol=2, byrow=TRUE))
```

```
##
##
## The following SaemixModel object was successfully created:
##
## Nonlinear mixed-effects model
##   Model function:  Count model Poisson
##   Model type:  likelihood
## function(psi,id,xidep) {
##   time<-xidep[,1]
##   y<-xidep[,2]
##   intercept<-psi[id,1]
##   slope<-psi[id,2]
##   lambda<- exp(intercept + slope*time)
##   logp <- -lambda + y*log(lambda) - log(factorial(y))
##   return(logp)
## }
##   Nb of parameters: 2
##       parameter names:  intercept slope
##       distribution:
##      Parameter Distribution Estimated
## [1,] intercept normal       Estimated
## [2,] slope     normal       Estimated
##   Variance-covariance matrix:
##           intercept slope
## intercept         1     1
## slope             1     1
##   Covariate model:
##      intercept slope
## [1,]         1     1
##     Initial values
##              intercept slope
## Pop.CondInit  1.609438  0.01
## Cov.CondInit  0.000000  0.00
```

```
saemix.options<-list(seed=632545,save=FALSE,save.graphs=FALSE, displayProgress=FALSE, fim=FALSE)

### Fit with saemix
poisson.fit<-saemix(saemix.model.poi,saemix.data,saemix.options)
```

```
## Nonlinear mixed-effects model fit by the SAEM algorithm
## -----------------------------------
## ----          Data             ----
## -----------------------------------
## Object of class SaemixData
##     longitudinal data for use with the SAEM algorithm
## Dataset rapi.saemix
##     Structured data: rapi ~ time + rapi | id
##     X variable for graphs: time (months)
##     covariates: gender ()
##       reference class for covariate gender :  Men
## Dataset characteristics:
##     number of subjects:     818
##     number of observations: 3616
##     average/min/max nb obs: 4.42  /  1  /  5
## First 10 lines of data:
##    id time rapi rapi.1 gender mdv cens occ ytype
## 1   1    0    0      0    Men   0    0   1     1
## 2   1    6    0      0    Men   0    0   1     1
## 3   1   18    0      0    Men   0    0   1     1
## 4   2    0    3      3  Women   0    0   1     1
## 5   2    6    6      6  Women   0    0   1     1
## 6   2   12    5      5  Women   0    0   1     1
## 7   2   18    4      4  Women   0    0   1     1
## 8   2   24    5      5  Women   0    0   1     1
## 9   3    0    9      9    Men   0    0   1     1
## 10  3   12    1      1    Men   0    0   1     1
## -----------------------------------
## ----          Model            ----
## -----------------------------------
## Nonlinear mixed-effects model
##   Model function:  Count model Poisson
##   Model type:  likelihood
## function(psi,id,xidep) {
##   time<-xidep[,1]
##   y<-xidep[,2]
##   intercept<-psi[id,1]
##   slope<-psi[id,2]
##   lambda<- exp(intercept + slope*time)
##   logp <- -lambda + y*log(lambda) - log(factorial(y))
##   return(logp)
## }
## <bytecode: 0x5583f4a7d460>
##   Nb of parameters: 2
##       parameter names:  intercept slope
##       distribution:
##      Parameter Distribution Estimated
## [1,] intercept normal       Estimated
## [2,] slope     normal       Estimated
##   Variance-covariance matrix:
##           intercept slope
## intercept         1     0
## slope             0     1
##     No covariate in the model.
##     Initial values
##              intercept slope
## Pop.CondInit  1.609438  0.01
## -----------------------------------
## ----    Key algorithm options  ----
## -----------------------------------
##     Estimation of individual parameters (MAP)
##     Estimation of log-likelihood by importance sampling
##     Number of iterations:  K1=300, K2=100
##     Number of chains:  1
##     Seed:  632545
##     Number of MCMC iterations for IS:  5000
##     Simulations:
##         nb of simulated datasets used for npde:  1000
##         nb of simulated datasets used for VPC:  100
##     Input/output
##         save the results to a file:  FALSE
##         save the graphs to files:  FALSE
## ----------------------------------------------------
## ----                  Results                   ----
## ----------------------------------------------------
## -----------------  Fixed effects  ------------------
## ----------------------------------------------------
##      Parameter Estimate
## [1,] intercept  1.577
## [2,] slope     -0.033
## ----------------------------------------------------
## -----------  Variance of random effects  -----------
## ----------------------------------------------------
##           Parameter        Estimate
## intercept omega2.intercept 0.9039
## slope     omega2.slope     0.0039
## ----------------------------------------------------
## ------  Correlation matrix of random effects  ------
## ----------------------------------------------------
##                  omega2.intercept omega2.slope
## omega2.intercept 1                0
## omega2.slope     0                1
## ----------------------------------------------------
## ---------------  Statistical criteria  -------------
## ----------------------------------------------------
##
```

```
## Likelihood computed by importance sampling
##       -2LL= 21486.75
##       AIC = 21496.75
##       BIC = 21520.29
## ----------------------------------------------------
```

```
poisson.fit.cov2<-saemix(saemix.model.poi.cov2,saemix.data,saemix.options)
```

```
## Nonlinear mixed-effects model fit by the SAEM algorithm
## -----------------------------------
## ----          Data             ----
## -----------------------------------
## Object of class SaemixData
##     longitudinal data for use with the SAEM algorithm
## Dataset rapi.saemix
##     Structured data: rapi ~ time + rapi | id
##     X variable for graphs: time (months)
##     covariates: gender ()
##       reference class for covariate gender :  Men
## Dataset characteristics:
##     number of subjects:     818
##     number of observations: 3616
##     average/min/max nb obs: 4.42  /  1  /  5
## First 10 lines of data:
##    id time rapi rapi.1 gender mdv cens occ ytype
## 1   1    0    0      0    Men   0    0   1     1
## 2   1    6    0      0    Men   0    0   1     1
## 3   1   18    0      0    Men   0    0   1     1
## 4   2    0    3      3  Women   0    0   1     1
## 5   2    6    6      6  Women   0    0   1     1
## 6   2   12    5      5  Women   0    0   1     1
## 7   2   18    4      4  Women   0    0   1     1
## 8   2   24    5      5  Women   0    0   1     1
## 9   3    0    9      9    Men   0    0   1     1
## 10  3   12    1      1    Men   0    0   1     1
## -----------------------------------
## ----          Model            ----
## -----------------------------------
## Nonlinear mixed-effects model
##   Model function:  Count model Poisson
##   Model type:  likelihood
## function(psi,id,xidep) {
##   time<-xidep[,1]
##   y<-xidep[,2]
##   intercept<-psi[id,1]
##   slope<-psi[id,2]
##   lambda<- exp(intercept + slope*time)
##   logp <- -lambda + y*log(lambda) - log(factorial(y))
##   return(logp)
## }
## <bytecode: 0x5583f4a7d460>
##   Nb of parameters: 2
##       parameter names:  intercept slope
##       distribution:
##      Parameter Distribution Estimated
## [1,] intercept normal       Estimated
## [2,] slope     normal       Estimated
##   Variance-covariance matrix:
##           intercept slope
## intercept         1     1
## slope             1     1
##   Covariate model:
##         [,1] [,2]
## gender    1    1
##     Initial values
##              intercept slope
## Pop.CondInit  1.609438  0.01
## Cov.CondInit  0.000000  0.00
## -----------------------------------
## ----    Key algorithm options  ----
## -----------------------------------
##     Estimation of individual parameters (MAP)
##     Estimation of log-likelihood by importance sampling
##     Number of iterations:  K1=300, K2=100
##     Number of chains:  1
##     Seed:  632545
##     Number of MCMC iterations for IS:  5000
##     Simulations:
##         nb of simulated datasets used for npde:  1000
##         nb of simulated datasets used for VPC:  100
##     Input/output
##         save the results to a file:  FALSE
##         save the graphs to files:  FALSE
## ----------------------------------------------------
## ----                 Results                   ----
## ----------------------------------------------------
## -----------------  Fixed effects  ------------------
## ----------------------------------------------------
##      Parameter              Estimate
## [1,] intercept               1.683
## [2,] beta_gender(intercept) -0.196
## [3,] slope                  -0.022
## [4,] beta_gender(slope)     -0.017
## ----------------------------------------------------
## -----------  Variance of random effects  -----------
## ----------------------------------------------------
##           Parameter        Estimate
## intercept omega2.intercept 0.9179
## slope     omega2.slope     0.0039
## ----------------------------------------------------
## ------  Correlation matrix of random effects  ------
## ----------------------------------------------------
##                  omega2.intercept omega2.slope
## omega2.intercept  1.00           -0.14
## omega2.slope     -0.14            1.00
## ----------------------------------------------------
## ---------------  Statistical criteria  -------------
## ----------------------------------------------------
##
```

```
## Likelihood computed by importance sampling
##       -2LL= 21454.94
##       AIC = 21470.94
##       BIC = 21508.59
## ----------------------------------------------------
```

```
### Results
if(FALSE) {
  cat("Poisson parameter at time 0 in base model: lambda_0=", exp(poisson.fit@results@fixed.effects[1])
  cat("Poisson parameter at time 24 in base model: lambda_24=", exp(poisson.fit@results@fixed.effects[1
}

# print(exp(poisson.fit@results@fixed.effects))
# exp(poisson.fit.cov2@results@fixed.effects)
```

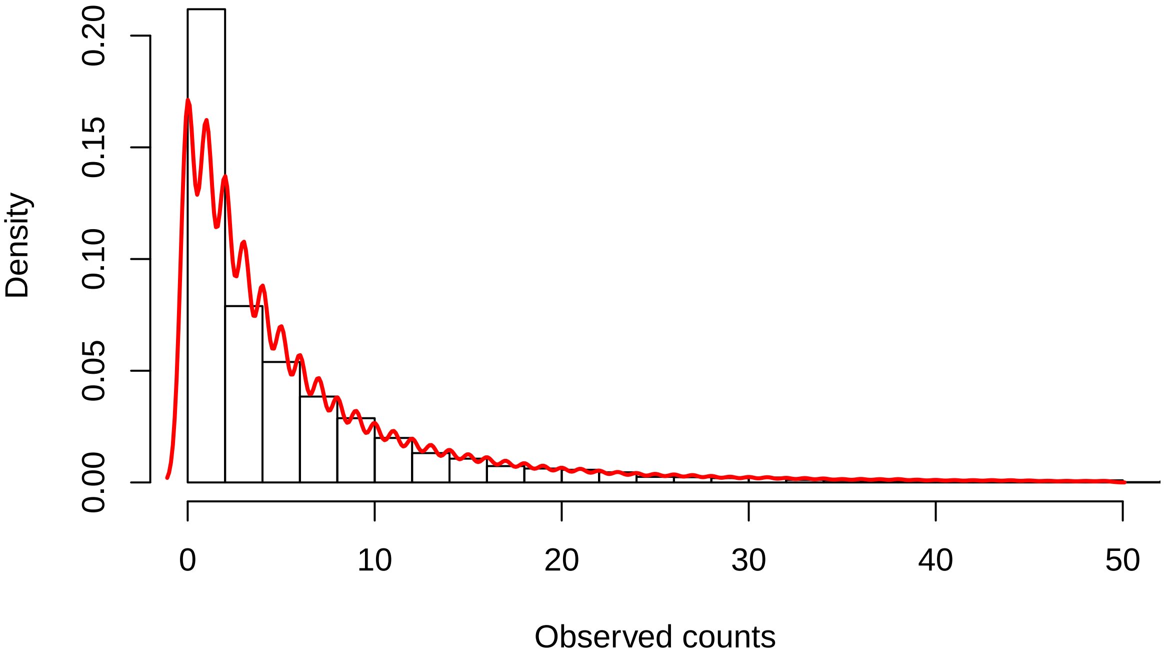
- Results
  - numerical output
    - the population value of the Poisson parameter at baseline is 4.8
    - we see a decreasing trend with time, with the population value of λ after 2 years decreasing to 2.2
  - convergence plots show good convergence for all parameters

Diagnostics

- Simulation function to simulate from a count model
  - the model function defines directly the log-pdf, so the user needs to define a function to simulate from the appropriate function
  - note the similarities between the model function (*count.poisson()*) and the simulation function (*countsimulate.poisson()*)
    - same setting of dependent variables (*time* and *rapi*) from *xidep* and parameters (*inter* and *slope*) from *psi*
      - note that we don't use *rapi* in *countsimulate.poisson()*
    - same definition of pevent ($=P(Y_{ij} = k)$, the probability of observing k counts)
    - in *count.poisson()* we then compute the probability of the observed outcome using the observed value of Y_{ij} contained in *rapi* for each observation
    - in *countsimulate.poisson()*, we use the individual value of $\lambda(Y_{ij})$ to simulate from a Poisson distribution using the *rpoisson()* function
- once the simulation function has been defined, we use the *simulateDiscreteSaemix()* function from the {saemix} package to simulate *nsim* values (here 100) with the population parameters estimated in *poisson.fit*
  - this adds a *simdata* element to the *poisson.fit*
  - we extract dataframe with the simulated data (*poisson.fit@sim.data@datasim*) and add a column *gender* to stratify the plots

```
### Simulations
nsim<-100
yfit1<-simulateDiscreteSaemix(poisson.fit.cov2, nsim=nsim)

hist(yfit1@data@data$rapi, xlim=c(0,50), freq=F, breaks=30, xlab="Observed counts", main="")
lines(density(yfit1@sim.data@datasim$ysim[yfit1@sim.data@datasim$ysim<50]), lwd = 2, col = 'red')
```



```
cat("Observed proportion of 0's", length(yfit1@data@data$rapi[yfit1@data@data$rapi==0])/yfit1@data@ntot
```

```
## Observed proportion of 0's 0.2090708
```

```
cat("      Poisson model, p=",length(yfit1@sim.data@datasim$ysim[yfit1@sim.data@datasim$ysim==0])/lengt
```

```
##       Poisson model, p= 0.1518501
```

```
# Checking proportion of zeroes
yfit<-yfit1
simdat <-yfit@sim.data@datasim
simdat$time<-rep(yfit@data@data$time,nsim)
simdat$gender<-rep(yfit@data@data$gender,nsim)

ytab<-NULL
for(irep in 1:nsim) {
  xtab<-simdat[simdat$irep==irep,]
  suppressMessages(
  xtab1 <- xtab %>%
    group_by(time, gender) %>%
    summarise(nev = sum(ysim==0), n=n()) %>%
    mutate(freq = nev/n)
  )
  ytab<-rbind(ytab,xtab1[,c("time","gender","freq")])
}
gtab <- ytab %>%
    group_by(time, gender) %>%
  summarise(lower=quantile(freq, c(0.05)), median=quantile(freq, c(0.5)), upper=quantile(freq, c(0.95))
  mutate(gender=ifelse(gender==0,"Men","Women"))
```

```
## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.
```

```
gtab$freq<-1
gtab1<-cbind(gtab, model="Poisson")
```

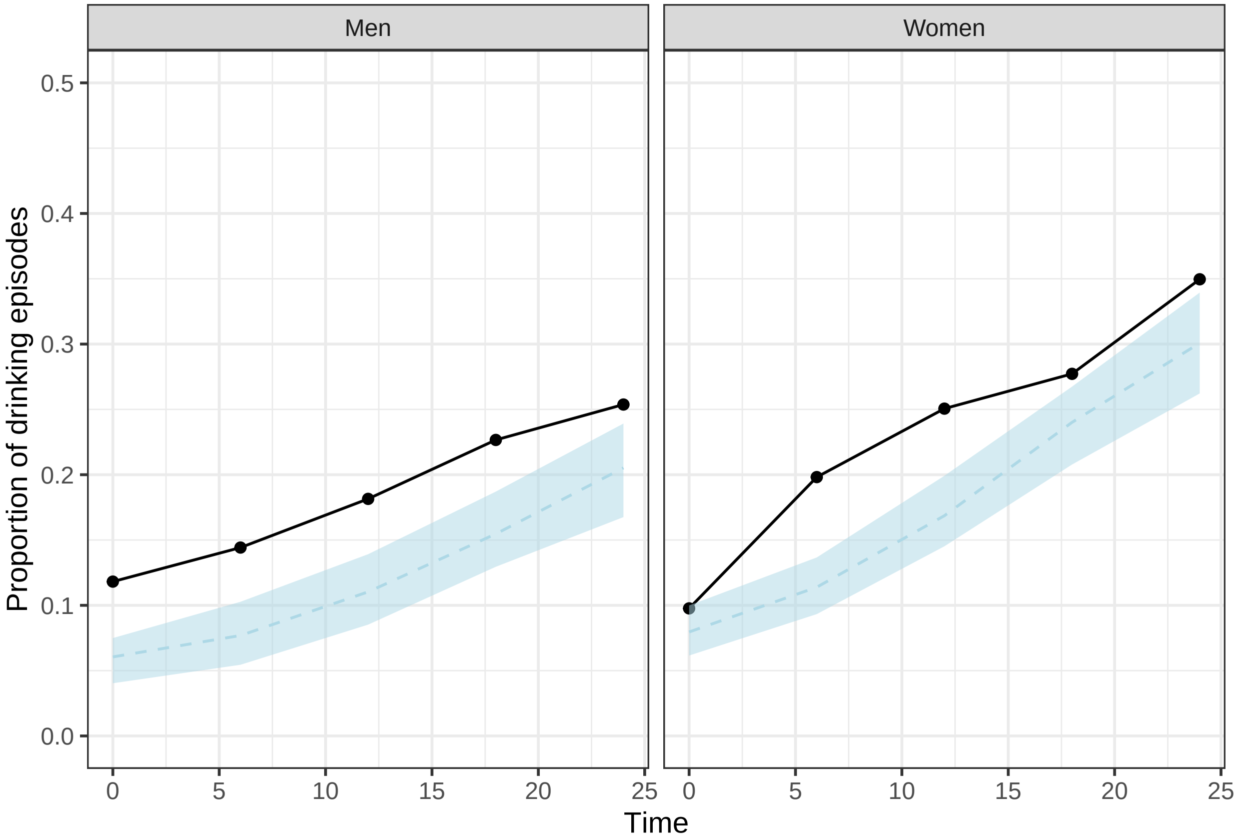
```r
rapipl <- rapi.saemix %>%
    group_by(time, gender) %>%
  summarise(nev = sum(rapi==0), n=n()) %>%
  mutate(freq = nev/n, sd=sqrt((1-nev/n)/nev)) %>%
  mutate(lower=freq-1.96*sd, upper=freq+1.96*sd)
```

```
## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.
```

```r
rapipl$lower[rapipl$lower<0] <-0 # we should use a better approximation for CI

plot2 <- ggplot(rapipl, aes(x=time, y=freq, group=gender)) + geom_line() +
  geom_point() +
  geom_line(data=gtab, aes(x=time, y=median,  group=gender), linetype=2, colour='lightblue') +
  geom_ribbon(data=gtab,aes(ymin=lower, ymax=upper,  group=gender), alpha=0.5, fill='lightblue') +
  ylim(c(0,0.5)) + theme_bw() + theme(legend.position = "none") + facet_wrap(.~gender) +
  xlab("Time") + ylab("Proportion of drinking episodes")

print(plot2)
```



Dealing with overdispersion The Poisson model in the previous section predicts a lower proportion of subjects without alcohol-related problems than we observe in data, a sign of overdispersion (with a Poisson model, the mean of the Poisson distribution, λ, is equal to the variance, an assumption which is violated here). Several models can be used to take this feature into account. First, we can use the Zero-Inflated Poisson model, where the number of counts equal to 0 is increased. This model can be built as a mixture between a distribution of 0's with probability p_0 and a standard Poisson model. No variability is set on p_0 which represents a proportion at the level of the population and we use a logit-normal distribution for this parameter to ensure it remains between 0 and 1.

```
## ZIP base model
saemix.model.zip<-saemixModel(model=count.poissonzip,description="count model ZIP",modeltype="likelihoo
                              psi0=matrix(c(1.5, 0.01, 0.2),ncol=3,byrow=TRUE,dimnames=list(NULL, c("in
                              transform.par=c(0,0,3), covariance.model=diag(c(1,1,0)), omega.init=diag(
```

```
##
##
## The following SaemixModel object was successfully created:
##
## Nonlinear mixed-effects model
##   Model function:  count model ZIP
##   Model type:  likelihood
## function(psi,id,xidep) {
##   time<-xidep[,1]
##   y<-xidep[,2]
##   intercept<-psi[id,1]
##   slope<-psi[id,2]
##   p0<-psi[id,3] # Probability of zero's
##   lambda<- exp(intercept + slope*time)
##   logp <- log(1-p0) -lambda + y*log(lambda) - log(factorial(y)) # Poisson
##   logp0 <- log(p0+(1-p0)*exp(-lambda)) # Zeroes
##   logp[y==0]<-logp0[y==0]
##   return(logp)
## }
##   Nb of parameters: 3
##       parameter names:  intercept slope p0
##       distribution:
##      Parameter Distribution Estimated
## [1,] intercept normal       Estimated
## [2,] slope     normal       Estimated
## [3,] p0        logit        Estimated
##   Variance-covariance matrix:
##           intercept slope p0
## intercept         1     0  0
## slope             0     1  0
## p0                0     0  0
##     No covariate in the model.
##     Initial values
##              intercept slope  p0
## Pop.CondInit       1.5  0.01 0.2
```

```
### ZIP Poisson with gender on both intercept
saemix.model.zip.cov1<-saemixModel(model=count.poissonzip,description="count model ZIP",modeltype="likel
                                   psi0=matrix(c(1.5, 0.01, 0.2),ncol=3,byrow=TRUE,dimnames=list(NULL, 
                                   transform.par=c(0,0,3), covariance.model=diag(c(1,1,0)), omega.init=
                                   covariate.model = matrix(c(1,0,0),ncol=3, byrow=TRUE))
```

```
##
##
## The following SaemixModel object was successfully created:
##
## Nonlinear mixed-effects model
##   Model function:  count model ZIP
##   Model type:  likelihood
## function(psi,id,xidep) {
##   time<-xidep[,1]
##   y<-xidep[,2]
##   intercept<-psi[id,1]
##   slope<-psi[id,2]
##   p0<-psi[id,3] # Probability of zero's
##   lambda<- exp(intercept + slope*time)
##   logp <- log(1-p0) -lambda + y*log(lambda) - log(factorial(y)) # Poisson
##   logp0 <- log(p0+(1-p0)*exp(-lambda)) # Zeroes
##   logp[y==0]<-logp0[y==0]
##   return(logp)
## }
##   Nb of parameters: 3
##       parameter names:  intercept slope p0
##       distribution:
##      Parameter Distribution Estimated
## [1,] intercept normal       Estimated
## [2,] slope     normal       Estimated
## [3,] p0        logit        Estimated
##   Variance-covariance matrix:
##           intercept slope p0
## intercept         1     0  0
## slope             0     1  0
## p0                0     0  0
##   Covariate model:
##      intercept slope p0
## [1,]         1     0  0
##     Initial values
##              intercept slope  p0
## Pop.CondInit       1.5  0.01 0.2
## Cov.CondInit       0.0  0.00 0.0
```

```
### ZIP Poisson with gender on both intercept and slope
saemix.model.zip.cov2<-saemixModel(model=count.poissonzip,description="count model ZIP",modeltype="likel
                                   psi0=matrix(c(1.5, 0.01, 0.2),ncol=3,byrow=TRUE,dimnames=list(NULL, 
                                   transform.par=c(0,0,3), covariance.model=diag(c(1,1,0)), omega.init=
                                   covariate.model = matrix(c(1,1,0),ncol=3, byrow=TRUE))
```

```
##
##
## The following SaemixModel object was successfully created:
##
## Nonlinear mixed-effects model
##   Model function:  count model ZIP
##   Model type:  likelihood
## function(psi,id,xidep) {
##   time<-xidep[,1]
##   y<-xidep[,2]
##   intercept<-psi[id,1]
##   slope<-psi[id,2]
##   p0<-psi[id,3] # Probability of zero's
##   lambda<- exp(intercept + slope*time)
##   logp <- log(1-p0) -lambda + y*log(lambda) - log(factorial(y)) # Poisson
##   logp0 <- log(p0+(1-p0)*exp(-lambda)) # Zeroes
##   logp[y==0]<-logp0[y==0]
##   return(logp)
## }
##   Nb of parameters: 3
##       parameter names:  intercept slope p0
##       distribution:
##      Parameter Distribution Estimated
## [1,] intercept normal       Estimated
## [2,] slope     normal       Estimated
## [3,] p0        logit        Estimated
##   Variance-covariance matrix:
##           intercept slope p0
## intercept         1     0  0
## slope             0     1  0
## p0                0     0  0
##   Covariate model:
##      intercept slope p0
## [1,]         1     1  0
##     Initial values
##               intercept slope  p0
## Pop.CondInit       1.5  0.01 0.2
## Cov.CondInit       0.0  0.00 0.0
```

```
zippoisson.fit<-saemix(saemix.model.zip,saemix.data,saemix.options)
```

```
## Nonlinear mixed-effects model fit by the SAEM algorithm
## -----------------------------------
## ----          Data             ----
## -----------------------------------
## Object of class SaemixData
##     longitudinal data for use with the SAEM algorithm
## Dataset rapi.saemix
##     Structured data: rapi ~ time + rapi | id
##     X variable for graphs: time (months)
##     covariates: gender ()
##       reference class for covariate gender :  Men
## Dataset characteristics:
##     number of subjects:     818
##     number of observations: 3616
##     average/min/max nb obs: 4.42  /  1  /  5
## First 10 lines of data:
##    id time rapi rapi.1 gender mdv cens occ ytype
## 1   1    0    0      0    Men   0    0   1     1
## 2   1    6    0      0    Men   0    0   1     1
## 3   1   18    0      0    Men   0    0   1     1
## 4   2    0    3      3  Women   0    0   1     1
## 5   2    6    6      6  Women   0    0   1     1
## 6   2   12    5      5  Women   0    0   1     1
## 7   2   18    4      4  Women   0    0   1     1
## 8   2   24    5      5  Women   0    0   1     1
## 9   3    0    9      9    Men   0    0   1     1
## 10  3   12    1      1    Men   0    0   1     1
## -----------------------------------
## ----          Model            ----
## -----------------------------------
## Nonlinear mixed-effects model
##   Model function:  count model ZIP
##   Model type:  likelihood
## function(psi,id,xidep) {
##   time<-xidep[,1]
##   y<-xidep[,2]
##   intercept<-psi[id,1]
##   slope<-psi[id,2]
##   p0<-psi[id,3] # Probability of zero's
##   lambda<- exp(intercept + slope*time)
##   logp <- log(1-p0) -lambda + y*log(lambda) - log(factorial(y)) # Poisson
##   logp0 <- log(p0+(1-p0)*exp(-lambda)) # Zeroes
##   logp[y==0]<-logp0[y==0]
##   return(logp)
## }
## <bytecode: 0x5583f6cf6130>
##   Nb of parameters: 3
##       parameter names:  intercept slope p0
##       distribution:
##      Parameter Distribution Estimated
## [1,] intercept normal       Estimated
## [2,] slope     normal       Estimated
## [3,] p0        logit        Estimated
##   Variance-covariance matrix:
##           intercept slope p0
## intercept         1     0  0
## slope             0     1  0
## p0                0     0  0
##     No covariate in the model.
##     Initial values
##              intercept slope  p0
## Pop.CondInit       1.5  0.01 0.2
## ----------------------------------
## ----    Key algorithm options  ----
## ----------------------------------
##     Estimation of individual parameters (MAP)
##     Estimation of log-likelihood by importance sampling
##     Number of iterations:  K1=300, K2=100
##     Number of chains:  1
##     Seed:  632545
##     Number of MCMC iterations for IS:  5000
##     Simulations:
##         nb of simulated datasets used for npde:  1000
##         nb of simulated datasets used for VPC:  100
##     Input/output
##         save the results to a file:  FALSE
##         save the graphs to files:  FALSE
## ----------------------------------------------------
## ----                  Results                   ----
## ----------------------------------------------------
## -----------------  Fixed effects  ------------------
## ----------------------------------------------------
##      Parameter Estimate
## [1,] intercept  1.657
## [2,] slope     -0.029
## [3,] p0         0.076
## ----------------------------------------------------
## -----------  Variance of random effects  -----------
## ----------------------------------------------------
##           Parameter        Estimate
## intercept omega2.intercept 0.7977
## slope     omega2.slope     0.0032
## ----------------------------------------------------
## ------  Correlation matrix of random effects  ------
## ----------------------------------------------------
##                  omega2.intercept omega2.slope
## omega2.intercept 1                0
## omega2.slope     0                1
## ----------------------------------------------------
## ---------------  Statistical criteria  -------------
## ----------------------------------------------------
##
## Likelihood computed by importance sampling
##       -2LL= 20479.88
##       AIC = 20491.88
##       BIC = 20520.12
## ----------------------------------------------------
```

```
zippoisson.fit.cov1<-saemix(saemix.model.zip.cov1,saemix.data,saemix.options)
```

```
## Nonlinear mixed-effects model fit by the SAEM algorithm
## -----------------------------------
## ----          Data             ----
## -----------------------------------
## Object of class SaemixData
##     longitudinal data for use with the SAEM algorithm
## Dataset rapi.saemix
##     Structured data: rapi ~ time + rapi | id
##     X variable for graphs: time (months)
##     covariates: gender ()
##       reference class for covariate gender :  Men
## Dataset characteristics:
##     number of subjects:     818
##     number of observations: 3616
##     average/min/max nb obs: 4.42  /  1  /  5
## First 10 lines of data:
##    id time rapi rapi.1 gender mdv cens occ ytype
## 1   1    0    0      0    Men   0    0   1     1
## 2   1    6    0      0    Men   0    0   1     1
## 3   1   18    0      0    Men   0    0   1     1
## 4   2    0    3      3  Women   0    0   1     1
## 5   2    6    6      6  Women   0    0   1     1
## 6   2   12    5      5  Women   0    0   1     1
## 7   2   18    4      4  Women   0    0   1     1
## 8   2   24    5      5  Women   0    0   1     1
## 9   3    0    9      9    Men   0    0   1     1
## 10  3   12    1      1    Men   0    0   1     1
## -----------------------------------
## ----          Model            ----
## -----------------------------------
## Nonlinear mixed-effects model
##   Model function:  count model ZIP
##   Model type:  likelihood
## function(psi,id,xidep) {
##   time<-xidep[,1]
##   y<-xidep[,2]
##   intercept<-psi[id,1]
##   slope<-psi[id,2]
##   p0<-psi[id,3] # Probability of zero's
##   lambda<- exp(intercept + slope*time)
##   logp <- log(1-p0) -lambda + y*log(lambda) - log(factorial(y)) # Poisson
##   logp0 <- log(p0+(1-p0)*exp(-lambda)) # Zeroes
##   logp[y==0]<-logp0[y==0]
##   return(logp)
## }
## <bytecode: 0x5583f6cf6130>
##   Nb of parameters: 3
##       parameter names:  intercept slope p0
##       distribution:
##      Parameter Distribution Estimated
## [1,] intercept normal       Estimated
## [2,] slope     normal       Estimated
## [3,] p0        logit        Estimated
##   Variance-covariance matrix:
##           intercept slope p0
## intercept         1     0  0
## slope             0     1  0
## p0                0     0  0
##   Covariate model:
##        [,1] [,2] [,3]
## gender    1    0    0
##     Initial values
##              intercept slope  p0
## Pop.CondInit       1.5  0.01 0.2
## Cov.CondInit       0.0  0.00 0.0
## -----------------------------------
## ----    Key algorithm options  ----
## -----------------------------------
##     Estimation of individual parameters (MAP)
##     Estimation of log-likelihood by importance sampling
##     Number of iterations:  K1=300, K2=100
##     Number of chains:  1
##     Seed:  632545
##     Number of MCMC iterations for IS:  5000
##     Simulations:
##         nb of simulated datasets used for npde:  1000
##         nb of simulated datasets used for VPC:  100
##     Input/output
##         save the results to a file:  FALSE
##         save the graphs to files:  FALSE
## ----------------------------------------------------
## ----                  Results                   ----
## ----------------------------------------------------
## -----------------  Fixed effects  ------------------
## ----------------------------------------------------
##      Parameter              Estimate
## [1,] intercept               1.786
## [2,] beta_gender(intercept) -0.226
## [3,] slope                  -0.029
## [4,] p0                      0.076
## ----------------------------------------------------
## -----------  Variance of random effects  -----------
## ----------------------------------------------------
##           Parameter        Estimate
## intercept omega2.intercept 0.7849
## slope     omega2.slope     0.0033
## ----------------------------------------------------
## ------  Correlation matrix of random effects  ------
## ----------------------------------------------------
##                  omega2.intercept omega2.slope
## omega2.intercept 1                0
## omega2.slope     0                1
## ----------------------------------------------------
## ---------------  Statistical criteria  -------------
## ----------------------------------------------------
##
## Likelihood computed by importance sampling
##       -2LL= 20469.41
##       AIC = 20483.41
##       BIC = 20516.35
## ----------------------------------------------------
```

```
zippoisson.fit.cov2<-saemix(saemix.model.zip.cov2,saemix.data,saemix.options)
```

```
## Nonlinear mixed-effects model fit by the SAEM algorithm
## -----------------------------------
## ----          Data             ----
## -----------------------------------
## Object of class SaemixData
##     longitudinal data for use with the SAEM algorithm
## Dataset rapi.saemix
##     Structured data: rapi ~ time + rapi | id
##     X variable for graphs: time (months)
##     covariates: gender ()
##       reference class for covariate gender :  Men
## Dataset characteristics:
##     number of subjects:     818
##     number of observations: 3616
##     average/min/max nb obs: 4.42  /  1  /  5
## First 10 lines of data:
##    id time rapi rapi.1 gender mdv cens occ ytype
## 1   1    0    0      0    Men   0    0   1     1
## 2   1    6    0      0    Men   0    0   1     1
## 3   1   18    0      0    Men   0    0   1     1
## 4   2    0    3      3  Women   0    0   1     1
## 5   2    6    6      6  Women   0    0   1     1
## 6   2   12    5      5  Women   0    0   1     1
## 7   2   18    4      4  Women   0    0   1     1
## 8   2   24    5      5  Women   0    0   1     1
## 9   3    0    9      9    Men   0    0   1     1
## 10  3   12    1      1    Men   0    0   1     1
## -----------------------------------
## ----          Model            ----
## -----------------------------------
## Nonlinear mixed-effects model
##   Model function:  count model ZIP
##   Model type:  likelihood
## function(psi,id,xidep) {
##   time<-xidep[,1]
##   y<-xidep[,2]
##   intercept<-psi[id,1]
##   slope<-psi[id,2]
##   p0<-psi[id,3] # Probability of zero's
##   lambda<- exp(intercept + slope*time)
##   logp <- log(1-p0) -lambda + y*log(lambda) - log(factorial(y)) # Poisson
##   logp0 <- log(p0+(1-p0)*exp(-lambda)) # Zeroes
##   logp[y==0]<-logp0[y==0]
##   return(logp)
## }
## <bytecode: 0x5583f6cf6130>
##   Nb of parameters: 3
##       parameter names:  intercept slope p0
##       distribution:
##      Parameter Distribution Estimated
## [1,] intercept normal       Estimated
## [2,] slope     normal       Estimated
## [3,] p0        logit        Estimated
##   Variance-covariance matrix:
##           intercept slope p0
## intercept         1     0  0
## slope             0     1  0
## p0                0     0  0
##   Covariate model:
##        [,1] [,2] [,3]
## gender    1    1    0
##     Initial values
##              intercept slope  p0
## Pop.CondInit       1.5  0.01 0.2
## Cov.CondInit       0.0  0.00 0.0
## -----------------------------------
## ----    Key algorithm options  ----
## -----------------------------------
##     Estimation of individual parameters (MAP)
##     Estimation of log-likelihood by importance sampling
##     Number of iterations:  K1=300, K2=100
##     Number of chains:  1
##     Seed:  632545
##     Number of MCMC iterations for IS:  5000
##     Simulations:
##         nb of simulated datasets used for npde:  1000
##         nb of simulated datasets used for VPC:  100
##     Input/output
##         save the results to a file:  FALSE
##         save the graphs to files:  FALSE
## ----------------------------------------------------
## ----                  Results                   ----
## ----------------------------------------------------
## -----------------  Fixed effects  ------------------
## ----------------------------------------------------
##      Parameter              Estimate
## [1,] intercept               1.773
## [2,] beta_gender(intercept) -0.197
## [3,] slope                  -0.020
## [4,] beta_gender(slope)     -0.016
## [5,] p0                      0.075
## ----------------------------------------------------
## -----------  Variance of random effects  -----------
## ----------------------------------------------------
##           Parameter        Estimate
## intercept omega2.intercept 0.7826
## slope     omega2.slope     0.0033
## ----------------------------------------------------
## ------  Correlation matrix of random effects  ------
## ----------------------------------------------------
##                  omega2.intercept omega2.slope
## omega2.intercept 1                0
## omega2.slope     0                1
## ----------------------------------------------------
## ---------------  Statistical criteria  -------------
## ----------------------------------------------------
##
## Likelihood computed by importance sampling
##       -2LL= 20459.27
##       AIC = 20475.27
##       BIC = 20512.93
## ----------------------------------------------------
```

```
exp(zippoisson.fit@results@fixed.effects)
```

```
## [1] 5.2450012 0.9714983 1.0793068
```

```
exp(zippoisson.fit.cov1@results@fixed.effects)
```

```
## [1] 5.9656256 0.7975888 0.9714754 1.0793259
```

```
exp(zippoisson.fit.cov2@results@fixed.effects)
```

```
## [1] 5.8872267 0.8213610 0.9797720 0.9844017 1.0783237
```

References

Atkins D

Comets E, Rodrigues C, Jullien V, Ursino M (2021). Conditional non-parametric bootstrap for non-linear mixed effect models. *Pharmaceutical Research*, 38: 1057-66.

Neighbors (2010a)

Neighbors (2010b)

Ueckert S, Mentré F (2017). A new method for evaluation of the Fisher information matrix for discrete mixed effect models using Monte Carlo sampling and adaptive Gaussian quadrature. *Computational Statistics and Data Analysis*, 111: 203-19. 10.1016/j.csda.2016.10.011

White (1989)