> ########## ########## Simulation Diabetes Beispiel mit Random Effects mit dem runjags package

> ########## Die Working Directory muss auf Ihre Bedürfnisse angepasst werden

>

>

>

>

>

> # Teil Simulation mit JAGS ------------------------------------------------------------------

>

>

>

>

> ##### Clear data

> rm(list=ls())

>

>

>

> ##### load libraries

> library(rjags)

> library(coda)

> library(random)

> library(matrixStats) # zusätzl Paket, berechnet Median

> load.module("glm")

> load.module("lecuyer")

> load.module("dic")

>

>

>

> ##### Sichergehen richtiger Working directory

> setwd("C:/Users/IvanB/Desktop/Masterarbeit/Statistische Programme und Gibbs Sampler/Programm JAGS/Nachrechnen TSD2-Dokument/Nachrechnen mit runjags/Ex3 Diabetes")

>

>

>

> ##### Read the data into R.

> #data = read.table("Diabetes\_Data\_neu sortiert.txt", sep = "", header=F)

> data = as.matrix(read.table("Diabetes\_Data\_neu sortiert.txt", sep = "", header=T))

> head(data) # Shows the first six entries

time.. t..1. t..2. t..3. r..1. r..2. r..3. n..1. n..2. n..3. na..

[1,] 5.8 1 2 3 43 34 37 1081 2213 1102 3

[2,] 4.7 1 2 NA 29 20 NA 416 424 NA 2

[3,] 3.0 1 2 NA 140 118 NA 1631 1578 NA 2

[4,] 3.8 1 3 NA 75 86 NA 3272 3297 NA 2

[5,] 4.0 1 4 5 302 154 119 6766 3954 4096 3

[6,] 3.0 1 4 NA 176 136 NA 2511 2508 NA 2

> #data2 = as.data.frread.table("Diabetes\_Data\_Rest.txt")

> #head(data2) # Shows the first six entries

>

>

>

> ##### Values for simulation, prepare dat for JAGS (allocation values from data)

> ns <- nrow(data)

> #ns # check

> nt <- 6

> #nt # check

> na <- data[,11]

> #na # check

> r <- data[,5:7]

> #r # Check

> time <- data[,1]

> #time # Check

> t <- data[,2:4]

> #t # Check

> n <- data[,8:10]

> #n # Check

>

>

>

> ##### read in inits with chains

> # Anmerkung: da cloglog als Link: wurden die Inits von JAGS generiert.

> # Die manuellen Inits sind zwar drin, können aber rausgelassen werden

>

> inits1 <- list(d=c(NA,0,0,0,0,0),

+ sd=1,

+ mu=c(0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0),

+ A=0 ,

+ delta= structure(.Data= c(NA, 0, 0, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, 0, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA,

+ 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0,0, NA, 0, 0, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA,

+ NA, 0, NA), .Dim=c(22, 3)))

> # .RNG.name="base::Wichmann-Hill", .RNG.seed=1

>

> inits2 <- list(d=c(NA,-1,4,-1,2,3),

+ sd=3,

+ mu=c(1,1,0,1,0, 0,1,0,0,0, 1,1,0,0,0, 0,1,0,0,0, 1,1),

+ A=1 ,

+ delta= structure(.Data= c(NA, 0, 0, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, 0, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA,

+ 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0,0, NA, 0, 0, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA,

+ NA, 0, NA), .Dim=c(22, 3)))

>

> inits3 <- list(d=c(NA,1,4,-3,-2,3),

+

+ sd=4.5,

+ mu=c(1,1,0,1,0, 0,1,0,0,0, 1,1,0,-2,0, 0,1,0,-2,0, 1,1),

+ A=2 ,

+ delta= structure(.Data= c(NA, 0, 0, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, 0, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA,

+ 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0,0, NA, 0, 0, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA, NA, 0, NA,

+ NA, 0, NA), .Dim=c(22, 3)))

> #, .RNG.name="base::Wichmann-Hill", .RNG.seed=3

>

> all.inits <- list(inits1, inits2, inits3)

> # all.inits <- list(inits2)

>

>

>

> ##### define JAGS model within R

> cat("model{ # \*\*\* PROGRAM STARTS

+ for(i in 1:ns){ # LOOP THROUGH STUDIES

+ mu[i] ~ dnorm(0,.0001) # vague priors for all trial baselines

+ for (k in 1:na[i]) { # LOOP THROUGH ARMS

+ r[i,k] ~ dbin(p[i,k],n[i,k]) # Binomial likelihood

+ cloglog(p[i,k]) <- log(time[i]) + mu[i] + d[t[i,k]] - d[t[i,1]] # model for linear predictor

+ rhat[i,k] <- p[i,k] \* n[i,k] # expected value of the numerators

+ dev[i,k] <- 2 \* (r[i,k] \* (log(r[i,k])-log(rhat[i,k])) + (n[i,k]-r[i,k]) \* (log(n[i,k]-r[i,k]) - log(n[i,k]-rhat[i,k]))) #Deviance contribution

+ }

+ resdev[i] <- sum(dev[i,1:na[i]]) # summed residual deviance contribution for this trial

+ }

+ totresdev <- sum(resdev[]) #Total Residual Deviance

+ d[1]<-0 # treatment effect is zero for reference treatment

+ for (k in 2:nt){ d[k] ~ dnorm(0,.0001) } # vague priors for treatment effects

+ A ~ dnorm(-4.2,1.11)

+ for (k in 1:nt) {

+ cloglog(T[k]) <- log(3) + A + d[k]

+ }

+ } ",

+ file="Diabetes\_Fixed.txt")

>

>

>

> ##### Set up the JAGS model and settings

> jags.m <- run.jags(model="Diabetes\_Fixed.txt", monitor=c("d[2]", "d[3]", "d[4]", "d[5]", "d[6]", "T[1]", "T[2]", "T[3]", "T[4]", "T[5]", "T[6]" , "totresdev", "deviance", "pd", "dic" , "full.pd" ),

+ data=list("ns"=ns, "nt"=nt, "na"=na, "r"=r, "time"=time, "t"=t, "n"=n) , n.chains=3, burnin = 10000, sample = 20000, adapt = 5000)

Compiling rjags model...

Calling the simulation using the rjags method...

Adapting the model for 5000 iterations...

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100%

Burning in the model for 10000 iterations...

|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*| 100%

Running the model for 20000 iterations...

|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*| 100%

Extending 20000 iterations for pD/DIC estimates...

|\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*| 100%

Simulation complete

Calculating summary statistics...

Calculating the Gelman-Rubin statistic for 13 variables....

Finished running the simulation

Warning message:

No initial values were provided - JAGS will use the same initial values for all chains

>

>

>

> #### optional, falls nicht konvergiert:

> #jags.m <- autorun.jags(model="Diabetes\_Fixed.txt", monitor=c("d[2]", "d[3]", "d[4]", "d[5]", "d[6]", "T[1]", "T[2]", "T[3]", "T[4]", "T[5]", "T[6]", "totresdev", "deviance", "pd", "dic" , "full.pd" ),

> # data=list("ns"=ns, "nt"=nt, "na"=na, "r"=r, "time"=time, "t"=t, "n"=n) , n.chains=3, inits=all.inits, burnin = 10000, sample = 20000, adapt = 5000, max.time="1hr")

>

>

>

>

>

> # Ausgabe posteriore Werte und Berechnung DIC --------------------------

>

>

>

>

> print(jags.m)

JAGS model summary statistics from 60000 samples (chains = 3; adapt+burnin = 15000):

Lower95 Median Upper95 Mean SD Mode MCerr MC%ofSD SSeff AC.10 psrf

d[2] -0.36022 -0.24736 -0.14228 -0.24776 0.055707 -- 0.0010871 2 2626 0.36861 1.0004

d[3] -0.16778 -0.05704 0.049817 -0.057851 0.055416 -- 0.0012346 2.2 2015 0.48346 1.0011

d[4] -0.35768 -0.25322 -0.14965 -0.25396 0.053404 -- 0.0011958 2.2 1994 0.49593 1.0015

d[5] -0.46108 -0.35918 -0.25533 -0.35935 0.052845 -- 0.001071 2 2435 0.37287 1.0005

d[6] -0.5746 -0.45355 -0.33006 -0.45371 0.062985 -- 0.0012998 2.1 2348 0.40953 1.0007

T[1] 0.0016322 0.044078 0.19457 0.06529 0.066891 -- 0.00038503 0.6 30182 -0.0016014 1

T[2] 0.0015506 0.034594 0.15636 0.051835 0.054496 -- 0.00031378 0.6 30163 -0.00040268 1

T[3] 0.0016659 0.041617 0.1856 0.06193 0.063923 -- 0.00036791 0.6 30188 -0.00063561 1

T[4] 0.0013231 0.034324 0.15526 0.051542 0.054266 -- 0.00031224 0.6 30205 -0.00081669 1

T[5] 0.001284 0.030953 0.14036 0.046667 0.049659 -- 0.00028606 0.6 30135 -0.00079853 1

T[6] 0.0011069 0.028232 0.12907 0.042665 0.045695 -- 0.00026315 0.6 30152 -0.0005046 1

totresdev 64.843 77.513 92.601 78.151 7.2937 -- 0.053739 0.7 18421 0.016727 1

deviance 385.33 398 413.09 398.64 7.2937 -- 0.053739 0.7 18421 0.016727 1

Model fit assessment:

DIC = 425.5398 (range between chains: 425.4637 - 425.6222)

[PED not available from the stored object]

Estimated effective number of parameters: pD = 26.89092

Total time taken: 1 minutes

> summary(jags.m$mcmc) # für 2.5 - 97.5 CrI

Iterations = 15001:35000

Thinning interval = 1

Number of chains = 3

Sample size per chain = 20000

1. Empirical mean and standard deviation for each variable,

plus standard error of the mean:

Mean SD Naive SE Time-series SE

d[2] -0.24786 0.05554 0.0002267 0.0010866

d[3] -0.05782 0.05546 0.0002264 0.0012168

d[4] -0.25397 0.05335 0.0002178 0.0011846

d[5] -0.35933 0.05268 0.0002151 0.0010568

d[6] -0.45358 0.06296 0.0002570 0.0013038

T[1] 0.06514 0.06692 0.0002732 0.0002749

T[2] 0.05173 0.05463 0.0002230 0.0002255

T[3] 0.06180 0.06397 0.0002612 0.0002639

T[4] 0.05143 0.05439 0.0002220 0.0002242

T[5] 0.04657 0.04980 0.0002033 0.0002054

T[6] 0.04259 0.04590 0.0001874 0.0001896

totresdev 78.18154 7.31185 0.0298505 0.0504513

deviance 398.66720 7.31185 0.0298505 0.0504513

2. Quantiles for each variable:

2.5% 25% 50% 75% 97.5%

d[2] -0.357250 -0.28527 -0.24740 -0.21023 -0.13964

d[3] -0.169095 -0.09455 -0.05694 -0.02017 0.04931

d[4] -0.360072 -0.29010 -0.25335 -0.21786 -0.15079

d[5] -0.463088 -0.39471 -0.35906 -0.32365 -0.25771

d[6] -0.576524 -0.49625 -0.45328 -0.41091 -0.33104

T[1] 0.006881 0.02322 0.04415 0.08223 0.25103

T[2] 0.005370 0.01814 0.03463 0.06480 0.20210

T[3] 0.006469 0.02187 0.04173 0.07784 0.23928

T[4] 0.005316 0.01803 0.03438 0.06446 0.20155

T[5] 0.004791 0.01626 0.03102 0.05812 0.18347

T[6] 0.004372 0.01480 0.02827 0.05318 0.16800

totresdev 65.770201 72.96369 77.52756 82.71690 94.28405

deviance 386.255861 393.44935 398.01322 403.20256 414.76971

>

>

>

>

>

> ########## ########## ########## Simulation beendet ########## ########## ##########