> ########## ########## Simulation Blocker Beispiel mit random effects mit STAN ########## ##########

>

>

>

> ##### Clear data

> rm(list=ls())

>

>

>

> #### Setting working directory

> setwd("C:/Users/IvanB/Desktop/Masterarbeit/Statistische Programme und Gibbs Sampler/STAN/Nachrechnen TSD2/Blocker")

>

>

> #### Requiering stan

> library("rstan")

> library("rstantools")

> rstan\_options(auto\_write = TRUE)

> #options(mc.cores = parallel::detectCores())

> Sys.setenv(LOCAL\_CPPFLAGS = '-march=native')

>

>

>

> #### Read in data

> data = read.csv2("Blocker\_Data\_neu sortiert\_VI.csv", header=TRUE, sep = ";", quote = "\"", dec = ",", fill = TRUE, comment.char = "")

> #data

>

>

>

> #### Assignment data to stan

> NO =nrow(data)

> NT=max(data$ï..Treatment\_t )

> NS=max(data$Studie)

> n=data$Gesamtanzahl\_n

> r=data$Erfolge\_r

> t=data$ï..Treatment\_t

> s=data$Studie

> base=data$ï..Treatment\_t

>

> data\_list <- list(NO=NO, NT=NT, NS=NS, n=n, r=r, t=t, s=s,base=base)

>

>

>

> #### Read in inits

> inits1 <- function(chain\_id = 1) {

+ list(d=c( NA, 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))

+ }

>

> inits2 <- function(chain\_id = 2) {

+ list(d=c( NA, -1),

+ sd=4,

+ mu=c(-3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3))

+ }

>

> inits3 <- function(chain\_id = 3) {

+ list(d=c( NA, 2),

+ sd=2,

+ mu=c(-3, 5, -1, -3, 7, -3, -4, -3, -3, 0, -3, -3,0, 3, 5, -3, -3, -1, -3, -7, -3, -3))

+ }

>

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

> #all.inits

>

>

>

> # Compiling

> m <- stan\_model('Model\_Random\_VIII\_final.stan')

hash mismatch so recompiling; make sure Stan code ends with a blank line

> m <- stan\_model('Model\_Random\_VIII\_final.stan')

>

> # Simulation

> stan\_samples <- sampling(m, data = data\_list, iter=40000, verbose=T, chain=4) # !! iter nachher erhöhen

CHECKING DATA AND PREPROCESSING FOR MODEL 'Model\_Random\_VIII\_final' NOW.

COMPILING MODEL 'Model\_Random\_VIII\_final' NOW.

STARTING SAMPLER FOR MODEL 'Model\_Random\_VIII\_final' NOW.

SAMPLING FOR MODEL 'Model\_Random\_VIII\_final' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 0 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.

Chain 1: Adjust your expectations accordingly!

Chain 1:

Chain 1:

Chain 1: Iteration: 1 / 40000 [ 0%] (Warmup)

Chain 1: Iteration: 4000 / 40000 [ 10%] (Warmup)

Chain 1: Iteration: 8000 / 40000 [ 20%] (Warmup)

Chain 1: Iteration: 12000 / 40000 [ 30%] (Warmup)

Chain 1: Iteration: 16000 / 40000 [ 40%] (Warmup)

Chain 1: Iteration: 20000 / 40000 [ 50%] (Warmup)

Chain 1: Iteration: 20001 / 40000 [ 50%] (Sampling)

Chain 1: Iteration: 24000 / 40000 [ 60%] (Sampling)

Chain 1: Iteration: 28000 / 40000 [ 70%] (Sampling)

Chain 1: Iteration: 32000 / 40000 [ 80%] (Sampling)

Chain 1: Iteration: 36000 / 40000 [ 90%] (Sampling)

Chain 1: Iteration: 40000 / 40000 [100%] (Sampling)

Chain 1:

Chain 1: Elapsed Time: 65.277 seconds (Warm-up)

Chain 1: 67.722 seconds (Sampling)

Chain 1: 132.999 seconds (Total)

Chain 1:

SAMPLING FOR MODEL 'Model\_Random\_VIII\_final' NOW (CHAIN 2).

Chain 2:

Chain 2: Gradient evaluation took 0 seconds

Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.

Chain 2: Adjust your expectations accordingly!

Chain 2:

Chain 2:

Chain 2: Iteration: 1 / 40000 [ 0%] (Warmup)

Chain 2: Iteration: 4000 / 40000 [ 10%] (Warmup)

Chain 2: Iteration: 8000 / 40000 [ 20%] (Warmup)

Chain 2: Iteration: 12000 / 40000 [ 30%] (Warmup)

Chain 2: Iteration: 16000 / 40000 [ 40%] (Warmup)

Chain 2: Iteration: 20000 / 40000 [ 50%] (Warmup)

Chain 2: Iteration: 20001 / 40000 [ 50%] (Sampling)

Chain 2: Iteration: 24000 / 40000 [ 60%] (Sampling)

Chain 2: Iteration: 28000 / 40000 [ 70%] (Sampling)

Chain 2: Iteration: 32000 / 40000 [ 80%] (Sampling)

Chain 2: Iteration: 36000 / 40000 [ 90%] (Sampling)

Chain 2: Iteration: 40000 / 40000 [100%] (Sampling)

Chain 2:

Chain 2: Elapsed Time: 65.48 seconds (Warm-up)

Chain 2: 67.31 seconds (Sampling)

Chain 2: 132.79 seconds (Total)

Chain 2:

SAMPLING FOR MODEL 'Model\_Random\_VIII\_final' NOW (CHAIN 3).

Chain 3:

Chain 3: Gradient evaluation took 0 seconds

Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.

Chain 3: Adjust your expectations accordingly!

Chain 3:

Chain 3:

Chain 3: Iteration: 1 / 40000 [ 0%] (Warmup)

Chain 3: Iteration: 4000 / 40000 [ 10%] (Warmup)

Chain 3: Iteration: 8000 / 40000 [ 20%] (Warmup)

Chain 3: Iteration: 12000 / 40000 [ 30%] (Warmup)

Chain 3: Iteration: 16000 / 40000 [ 40%] (Warmup)

Chain 3: Iteration: 20000 / 40000 [ 50%] (Warmup)

Chain 3: Iteration: 20001 / 40000 [ 50%] (Sampling)

Chain 3: Iteration: 24000 / 40000 [ 60%] (Sampling)

Chain 3: Iteration: 28000 / 40000 [ 70%] (Sampling)

Chain 3: Iteration: 32000 / 40000 [ 80%] (Sampling)

Chain 3: Iteration: 36000 / 40000 [ 90%] (Sampling)

Chain 3: Iteration: 40000 / 40000 [100%] (Sampling)

Chain 3:

Chain 3: Elapsed Time: 62.639 seconds (Warm-up)

Chain 3: 68.031 seconds (Sampling)

Chain 3: 130.67 seconds (Total)

Chain 3:

SAMPLING FOR MODEL 'Model\_Random\_VIII\_final' NOW (CHAIN 4).

Chain 4:

Chain 4: Gradient evaluation took 0 seconds

Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.

Chain 4: Adjust your expectations accordingly!

Chain 4:

Chain 4:

Chain 4: Iteration: 1 / 40000 [ 0%] (Warmup)

Chain 4: Iteration: 4000 / 40000 [ 10%] (Warmup)

Chain 4: Iteration: 8000 / 40000 [ 20%] (Warmup)

Chain 4: Iteration: 12000 / 40000 [ 30%] (Warmup)

Chain 4: Iteration: 16000 / 40000 [ 40%] (Warmup)

Chain 4: Iteration: 20000 / 40000 [ 50%] (Warmup)

Chain 4: Iteration: 20001 / 40000 [ 50%] (Sampling)

Chain 4: Iteration: 24000 / 40000 [ 60%] (Sampling)

Chain 4: Iteration: 28000 / 40000 [ 70%] (Sampling)

Chain 4: Iteration: 32000 / 40000 [ 80%] (Sampling)

Chain 4: Iteration: 36000 / 40000 [ 90%] (Sampling)

Chain 4: Iteration: 40000 / 40000 [100%] (Sampling)

Chain 4:

Chain 4: Elapsed Time: 66.122 seconds (Warm-up)

Chain 4: 64.932 seconds (Sampling)

Chain 4: 131.054 seconds (Total)

Chain 4:

>

>

>

> #### Plotting and summarizing the posterior distribution

> stan\_samples # = print(stan\_samples)

Inference for Stan model: Model\_Random\_VIII\_final.

4 chains, each with iter=40000; warmup=20000; thin=1;

post-warmup draws per chain=20000, total post-warmup draws=80000.

mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat

d[1] -0.01 0.00 2.13 -4.20 -1.43 -0.01 1.43 4.20 187204 1

d[2] 0.00 0.00 2.14 -4.21 -1.44 -0.01 1.43 4.20 193385 1

mu[1] -2.55 0.00 0.46 -3.53 -2.84 -2.52 -2.23 -1.71 136269 1

mu[2] -2.22 0.00 0.26 -2.74 -2.39 -2.21 -2.04 -1.72 131111 1

mu[3] -2.19 0.00 0.29 -2.78 -2.38 -2.18 -1.99 -1.64 147920 1

mu[4] -2.42 0.00 0.09 -2.60 -2.48 -2.42 -2.36 -2.24 120982 1

mu[5] -2.52 0.00 0.17 -2.88 -2.64 -2.52 -2.40 -2.19 129486 1

mu[6] -2.30 0.00 0.37 -3.07 -2.54 -2.29 -2.05 -1.62 143376 1

mu[7] -1.69 0.00 0.09 -1.86 -1.75 -1.69 -1.63 -1.52 110875 1

mu[8] -2.20 0.00 0.14 -2.47 -2.29 -2.19 -2.10 -1.93 121923 1

mu[9] -1.98 0.00 0.16 -2.31 -2.09 -1.98 -1.87 -1.67 118388 1

mu[10] -2.24 0.00 0.08 -2.39 -2.29 -2.24 -2.19 -2.10 117150 1

mu[11] -2.37 0.00 0.13 -2.63 -2.46 -2.37 -2.28 -2.11 120184 1

mu[12] -1.55 0.00 0.15 -1.84 -1.65 -1.55 -1.45 -1.27 127119 1

mu[13] -3.03 0.00 0.24 -3.51 -3.19 -3.02 -2.87 -2.59 130433 1

mu[14] -2.88 0.00 0.14 -3.16 -2.97 -2.88 -2.79 -2.62 118313 1

mu[15] -1.40 0.00 0.18 -1.77 -1.52 -1.40 -1.28 -1.05 126311 1

mu[16] -1.56 0.00 0.16 -1.88 -1.67 -1.56 -1.45 -1.25 123789 1

mu[17] -2.18 0.00 0.23 -2.65 -2.33 -2.17 -2.02 -1.73 116909 1

mu[18] -3.11 0.00 0.31 -3.75 -3.31 -3.10 -2.90 -2.54 133170 1

mu[19] -3.60 0.00 0.38 -4.40 -3.85 -3.58 -3.33 -2.90 119608 1

mu[20] -1.54 0.00 0.16 -1.86 -1.65 -1.54 -1.44 -1.24 124576 1

mu[21] -2.12 0.00 0.16 -2.43 -2.23 -2.12 -2.02 -1.82 110753 1

mu[22] -2.91 0.00 0.16 -3.23 -3.01 -2.91 -2.80 -2.60 119039 1

delta[1] 0.00 0.00 0.10 -0.19 -0.07 0.00 0.07 0.19 191670 1

delta[2] 0.00 0.00 0.10 -0.19 -0.07 0.00 0.07 0.20 189229 1

delta[3] 0.00 0.00 0.10 -0.19 -0.07 0.00 0.07 0.20 195783 1

delta[4] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 174242 1

delta[5] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 194948 1

delta[6] 0.00 0.00 0.10 -0.19 -0.07 0.00 0.07 0.20 179789 1

delta[7] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 190910 1

delta[8] 0.00 0.00 0.10 -0.19 -0.07 0.00 0.07 0.19 184584 1

delta[9] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 184332 1

delta[10] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.19 183466 1

delta[11] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 183576 1

delta[12] 0.00 0.00 0.10 -0.19 -0.07 0.00 0.07 0.20 190735 1

delta[13] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 194782 1

delta[14] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 185848 1

delta[15] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 182949 1

delta[16] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 192156 1

delta[17] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 188487 1

delta[18] 0.00 0.00 0.10 -0.19 -0.07 0.00 0.07 0.20 193155 1

delta[19] 0.00 0.00 0.10 -0.20 -0.07 0.00 0.07 0.20 190752 1

delta[20] 0.00 0.00 0.10 -0.19 -0.07 0.00 0.07 0.20 192378 1

delta[21] 0.00 0.00 0.10 -0.19 -0.07 0.00 0.07 0.20 192110 1

delta[22] 0.00 0.00 0.10 -0.19 -0.07 0.00 0.07 0.20 193262 1

delta[23] 0.00 0.00 0.30 -0.60 -0.19 0.00 0.19 0.61 134671 1

delta[24] -0.23 0.00 0.27 -0.81 -0.40 -0.21 -0.04 0.26 97050 1

delta[25] -0.14 0.00 0.28 -0.72 -0.31 -0.13 0.05 0.38 120856 1

delta[26] -0.20 0.00 0.13 -0.45 -0.29 -0.20 -0.11 0.04 112811 1

delta[27] 0.04 0.00 0.21 -0.37 -0.10 0.03 0.17 0.45 125756 1

delta[28] -0.11 0.00 0.29 -0.72 -0.29 -0.10 0.08 0.44 117154 1

delta[29] -0.42 0.00 0.13 -0.68 -0.51 -0.42 -0.33 -0.16 93277 1

delta[30] -0.05 0.00 0.17 -0.38 -0.17 -0.05 0.06 0.28 119023 1

delta[31] -0.23 0.00 0.21 -0.66 -0.37 -0.22 -0.09 0.16 107694 1

delta[32] -0.29 0.00 0.11 -0.51 -0.36 -0.29 -0.21 -0.07 108579 1

delta[33] -0.15 0.00 0.16 -0.48 -0.26 -0.15 -0.04 0.17 114426 1

delta[34] -0.02 0.00 0.18 -0.39 -0.15 -0.02 0.10 0.33 125404 1

delta[35] -0.21 0.00 0.26 -0.76 -0.37 -0.20 -0.04 0.26 103331 1

delta[36] 0.19 0.00 0.17 -0.14 0.07 0.18 0.30 0.54 107804 1

delta[37] -0.16 0.00 0.22 -0.61 -0.30 -0.16 -0.02 0.24 113502 1

delta[38] -0.08 0.00 0.20 -0.47 -0.21 -0.08 0.05 0.30 121900 1

delta[39] 0.06 0.00 0.23 -0.39 -0.09 0.06 0.21 0.54 109252 1

delta[40] 0.08 0.00 0.27 -0.45 -0.10 0.07 0.25 0.64 119288 1

delta[41] 0.07 0.00 0.29 -0.49 -0.11 0.07 0.25 0.67 116664 1

delta[42] -0.13 0.00 0.20 -0.52 -0.26 -0.12 0.01 0.25 118635 1

delta[43] -0.34 0.00 0.21 -0.77 -0.48 -0.34 -0.20 0.04 88897 1

delta[44] -0.34 0.00 0.21 -0.78 -0.48 -0.33 -0.19 0.06 87691 1

s\_d 0.11 0.00 0.06 0.03 0.06 0.09 0.13 0.27 37521 1

A[1] -2.20 0.00 0.30 -2.79 -2.40 -2.20 -1.99 -1.61 189623 1

A[2] -2.20 0.00 0.30 -2.79 -2.40 -2.20 -1.99 -1.61 189601 1

d\_II[1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

d\_II[2] 0.00 0.00 2.14 -4.21 -1.44 -0.01 1.43 4.20 193385 1

tau 0.31 0.00 0.09 0.17 0.25 0.30 0.36 0.52 34885 1

w[1,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[1,2] 0.00 0.00 0.30 -0.60 -0.19 0.00 0.19 0.61 134671 1

w[2,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[2,2] -0.23 0.00 0.27 -0.81 -0.40 -0.21 -0.04 0.26 97050 1

w[3,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[3,2] -0.14 0.00 0.28 -0.72 -0.31 -0.13 0.05 0.38 120856 1

w[4,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[4,2] -0.20 0.00 0.13 -0.45 -0.29 -0.20 -0.11 0.04 112811 1

w[5,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[5,2] 0.04 0.00 0.21 -0.37 -0.10 0.03 0.17 0.45 125756 1

w[6,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[6,2] -0.11 0.00 0.29 -0.72 -0.29 -0.10 0.08 0.44 117154 1

w[7,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[7,2] -0.42 0.00 0.13 -0.68 -0.51 -0.42 -0.33 -0.16 93277 1

w[8,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[8,2] -0.05 0.00 0.17 -0.38 -0.17 -0.05 0.06 0.28 119023 1

w[9,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[9,2] -0.23 0.00 0.21 -0.66 -0.37 -0.22 -0.09 0.16 107694 1

w[10,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[10,2] -0.29 0.00 0.11 -0.51 -0.36 -0.29 -0.21 -0.07 108579 1

w[11,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[11,2] -0.15 0.00 0.16 -0.48 -0.26 -0.15 -0.04 0.17 114426 1

w[12,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[12,2] -0.02 0.00 0.18 -0.39 -0.15 -0.02 0.10 0.33 125404 1

w[13,1] 0.00 NaN 0.00 0.00 0.00 0.00 0.00 0.00 NaN NaN

w[13,2] -0.21 0.00 0.26 -0.76 -0.37 -0.20 -0.04 0.26 103331 1

[ reached getOption("max.print") -- omitted 374 rows ]

Samples were drawn using NUTS(diag\_e) at Sun Aug 18 17:55:08 2019.

For each parameter, n\_eff is a crude measure of effective sample size,

and Rhat is the potential scale reduction factor on split chains (at

convergence, Rhat=1).

> plot(stan\_samples)

'pars' not specified. Showing first 10 parameters by default.

ci\_level: 0.8 (80% intervals)

outer\_level: 0.95 (95% intervals)

Warning message:

In doTryCatch(return(expr), name, parentenv, handler) :

display list redraw incomplete

> Stan\_summary <- summary(stan\_samples, pars = c("d[2]","d\_II[2]", "T[1]", "T[2]", "s\_d", "totalresdev"), probs = c(0.025, 0.975))$summary

> Stan\_summary

mean se\_mean sd 2.5% 97.5% n\_eff Rhat

d[2] -0.004338407 0.0048587863 2.13668035 -4.211332703 4.2021903 193385.32 0.9999629

d\_II[2] -0.004338407 0.0048587863 2.13668035 -4.211332703 4.2021903 193385.32 0.9999629

T[1] 0.103103374 0.0000695413 0.02816467 0.057609209 0.1668806 164030.15 0.9999777

T[2] 0.213005779 0.0008379001 0.25216138 0.001617101 0.8851220 90567.55 0.9999998

s\_d 0.105865829 0.0003268107 0.06330453 0.028693693 0.2654089 37521.22 1.0002477

totalresdev 41.231304559 0.0424575617 8.62179794 26.152554181 59.7559432 41236.86 1.0000311

>

>

>

> #### Additional Lines for Median

> Median\_d2 <- median(as.matrix(stan\_samples, pars = c("d\_II[2]")))

> Median\_d2

[1] -0.009859652

> Median\_T1 <- median(as.matrix(stan\_samples, pars = c("T[1]")))

> Median\_T1

[1] 0.09981241

> Median\_T2 <- median(as.matrix(stan\_samples, pars = c("T[2]")))

> Median\_T2

[1] 0.0998301

> Median\_sd <- median(as.matrix(stan\_samples, pars = c("s\_d")))

> Median\_sd

[1] 0.09156049

>

>

>

>

>

> # Section for Convergence Diagnostic --------------------------------------------------

>

>

> # konkreter Vergleich mit BUGS nicht möglich ->kein pD in dem Sinne

> # DIC (und damit pD) ist veraltet

> # => loo()-Fkt und WAIC

> #(pD Code für Python)

> #allerdings andere Konvergenz - Diagnostika:

> # Diagnostik mir rstan Paket

> sampler\_params <- get\_sampler\_params(stan\_samples, inc\_warmup = TRUE)

> summary(do.call(rbind, sampler\_params), digits = 2)

accept\_stat\_\_ stepsize\_\_ treedepth\_\_ n\_leapfrog\_\_ divergent\_\_ energy\_\_

Min. :0.00 Min. : 0.011 Min. :0.0 Min. : 1 Min. :0.00000 Min. : 5910

1st Qu.:0.76 1st Qu.: 0.323 1st Qu.:4.0 1st Qu.: 15 1st Qu.:0.00000 1st Qu.: 5936

Median :0.88 Median : 0.377 Median :4.0 Median : 15 Median :0.00000 Median : 5942

Mean :0.84 Mean : 0.377 Mean :3.8 Mean : 15 Mean :0.00074 Mean : 5943

3rd Qu.:0.96 3rd Qu.: 0.418 3rd Qu.:4.0 3rd Qu.: 15 3rd Qu.:0.00000 3rd Qu.: 5948

Max. :1.00 Max. :14.386 Max. :8.0 Max. :511 Max. :1.00000 Max. :17820

> # each chain separately

> lapply(sampler\_params, summary, digits = 2)

[[1]]

accept\_stat\_\_ stepsize\_\_ treedepth\_\_ n\_leapfrog\_\_ divergent\_\_ energy\_\_

Min. :0.00 Min. : 0.011 Min. :0.0 Min. : 1 Min. :0e+00 Min. : 5911

1st Qu.:0.77 1st Qu.: 0.327 1st Qu.:4.0 1st Qu.: 15 1st Qu.:0e+00 1st Qu.: 5937

Median :0.89 Median : 0.327 Median :4.0 Median : 15 Median :0e+00 Median : 5942

Mean :0.84 Mean : 0.374 Mean :3.8 Mean : 15 Mean :9e-04 Mean : 5943

3rd Qu.:0.96 3rd Qu.: 0.419 3rd Qu.:4.0 3rd Qu.: 15 3rd Qu.:0e+00 3rd Qu.: 5949

Max. :1.00 Max. :11.095 Max. :8.0 Max. :447 Max. :1e+00 Max. :12984

[[2]]

accept\_stat\_\_ stepsize\_\_ treedepth\_\_ n\_leapfrog\_\_ divergent\_\_ energy\_\_

Min. :0.00 Min. :0.012 Min. :0.0 Min. : 1 Min. :0e+00 Min. : 5912

1st Qu.:0.77 1st Qu.:0.323 1st Qu.:4.0 1st Qu.: 15 1st Qu.:0e+00 1st Qu.: 5936

Median :0.89 Median :0.323 Median :4.0 Median : 15 Median :0e+00 Median : 5942

Mean :0.84 Mean :0.371 Mean :3.8 Mean : 15 Mean :7e-04 Mean : 5943

3rd Qu.:0.96 3rd Qu.:0.418 3rd Qu.:4.0 3rd Qu.: 15 3rd Qu.:0e+00 3rd Qu.: 5948

Max. :1.00 Max. :7.847 Max. :8.0 Max. :511 Max. :1e+00 Max. :13090

[[3]]

accept\_stat\_\_ stepsize\_\_ treedepth\_\_ n\_leapfrog\_\_ divergent\_\_ energy\_\_

Min. :0.00 Min. : 0.011 Min. :0.0 Min. : 1 Min. :0.00000 Min. : 5910

1st Qu.:0.78 1st Qu.: 0.309 1st Qu.:4.0 1st Qu.: 15 1st Qu.:0.00000 1st Qu.: 5936

Median :0.89 Median : 0.309 Median :4.0 Median : 15 Median :0.00000 Median : 5942

Mean :0.85 Mean : 0.365 Mean :3.8 Mean : 15 Mean :0.00057 Mean : 5943

3rd Qu.:0.96 3rd Qu.: 0.420 3rd Qu.:4.0 3rd Qu.: 15 3rd Qu.:0.00000 3rd Qu.: 5949

Max. :1.00 Max. :14.386 Max. :8.0 Max. :415 Max. :1.00000 Max. :12411

[[4]]

accept\_stat\_\_ stepsize\_\_ treedepth\_\_ n\_leapfrog\_\_ divergent\_\_ energy\_\_

Min. :0.00 Min. : 0.012 Min. :0.0 Min. : 1 Min. :0.00000 Min. : 5910

1st Qu.:0.73 1st Qu.: 0.377 1st Qu.:3.0 1st Qu.: 15 1st Qu.:0.00000 1st Qu.: 5936

Median :0.86 Median : 0.377 Median :4.0 Median : 15 Median :0.00000 Median : 5942

Mean :0.82 Mean : 0.398 Mean :3.7 Mean : 15 Mean :0.00077 Mean : 5943

3rd Qu.:0.95 3rd Qu.: 0.417 3rd Qu.:4.0 3rd Qu.: 15 3rd Qu.:0.00000 3rd Qu.: 5948

Max. :1.00 Max. :12.743 Max. :8.0 Max. :319 Max. :1.00000 Max. :17820

> Stan\_summary\_lp\_\_ <- summary(stan\_samples, pars = c("lp\_\_"), probs = c(0.025, 0.975))$summary # sigmasq\_delta entspricht sd, nachher ändern

> Stan\_summary\_lp\_\_

mean se\_mean sd 2.5% 97.5% n\_eff Rhat

lp\_\_ -5907.153 0.04279102 6.69149 -5921.137 -5894.852 24453.46 1.000196

> # weitere Möglichkeit: Package 'shinystan'

>

>

>

>

>

> # Section for leverage plot -----------------------------------------------

>

>

>

> #### Read in single values for dev and rhat

>

> SingeValues\_dev <- summary(stan\_samples, pars = c("dev[1]", "dev[2]", "dev[3]", "dev[4]", "dev[5]", "dev[6]", "dev[7]", "dev[8]", "dev[9]", "dev[10]", "dev[11]", "dev[12]", "dev[13]", "dev[14]", "dev[15]", "dev[16]", "dev[17]", "dev[18]", "dev[19]", "dev[20]", "dev[21]", "dev[22]", "dev[23]", "dev[24]", "dev[25]", "dev[26]", "dev[27]", "dev[28]", "dev[29]", "dev[30]", "dev[31]", "dev[32]", "dev[33]", "dev[34]", "dev[35]", "dev[36]", "dev[37]", "dev[38]", "dev[39]", "dev[40]", "dev[41]", "dev[42]", "dev[43]", "dev[44]"

+ ))$summary

> #SingeValues\_dev

> SingeValues\_rhat <- summary(stan\_samples, pars = c("rhat[1]", "rhat[2]", "rhat[3]", "rhat[4]", "rhat[5]", "rhat[6]", "rhat[7]", "rhat[8]", "rhat[9]", "rhat[10]", "rhat[11]", "rhat[12]", "rhat[13]", "rhat[14]", "rhat[15]", "rhat[16]", "rhat[17]", "rhat[18]", "rhat[19]", "rhat[20]", "rhat[21]", "rhat[22]", "rhat[23]", "rhat[24]", "rhat[25]", "rhat[26]", "rhat[27]", "rhat[28]", "rhat[29]", "rhat[30]", "rhat[31]", "rhat[32]", "rhat[33]", "rhat[34]", "rhat[35]", "rhat[36]", "rhat[37]", "rhat[38]", "rhat[39]", "rhat[40]", "rhat[41]", "rhat[42]", "rhat[43]", "rhat[44]"

+ ))$summary

> #SingeValues\_rhat

>

> out\_lePlo <- capture.output( SingeValues\_dev)

> cat("Hilf\_lePlo", out\_lePlo, file="Hilf\_lePlo.txt", sep="\n", append=TRUE)

> out\_lePlo <- capture.output( SingeValues\_rhat)

> cat("Hilf\_lePlo", out\_lePlo, file="Hilf\_lePlo.txt", sep="\n", append=TRUE)

>

>

>

> Hilf\_dev\_I <- read.table("Hilf\_lePlo.txt", sep = "", header=F, skip=2, nrows=22)

> #Hilf\_dev\_I

> Hilf\_dev\_II <- read.table("Hilf\_lePlo.txt", sep = "", header=F, skip=24, nrows=22)

> #Hilf\_dev\_II

> Hilf\_rhat\_I <- read.table("Hilf\_lePlo.txt", sep = "", header=F, skip=48, nrows=22)

> #Hilf\_rhat\_I

> Hilf\_rhat\_II <- read.table("Hilf\_lePlo.txt", sep = "", header=F, skip=70, nrows=22)

> #Hilf\_rhat\_II

>

>

>

> #### Berechnung w\_ik

> Hilf\_dev\_1 <- cbind(Hilf\_dev\_I[,2], Hilf\_dev\_II[,2])

> #Hilf\_dev\_1

> Hilf\_dev\_2 <- cbind(Hilf\_dev\_1, total = rowMeans(Hilf\_dev\_1))

> #Hilf\_dev\_2

> w\_ik <- sqrt(Hilf\_dev\_2[,3])

> w\_ik\_neg <- -w\_ik

>

> fertige\_Daten\_w\_ik <- cbind(Hilf\_dev\_2, w\_ik\_neg, w\_ik)

> #fertige\_Daten\_w\_ik

>

>

>

> #### Berechnung leverage\_ik

> dev\_tilde\_erst\_I <- data[1:22,2]\*log(data[1:22,2]/Hilf\_rhat\_I[,2])

> #dev\_tilde\_erst\_I

> dev\_tilde\_zweit\_I <- (data[1:22,3]-data[1:22,2])\*log((data[1:22,3]-data[1:22,2])/(data[1:22,3]-Hilf\_rhat\_I[,2]))

> #dev\_tilde\_zweit\_I

> dev\_tilde\_gesamt\_I <- 2\*(dev\_tilde\_erst\_I+dev\_tilde\_zweit\_I)

> #dev\_tilde\_gesamt\_I

>

> dev\_tilde\_erst\_II <- data[23:44,2]\*log(data[23:44,2]/Hilf\_rhat\_II[,2])

> #dev\_tilde\_erst\_II

> dev\_tilde\_zweit\_II <- (data[23:44,3]-data[23:44,2])\*log((data[23:44,3]-data[23:44,2])/(data[23:44,3]-Hilf\_rhat\_II[,2]))

> #dev\_tilde\_zweit\_II

> dev\_tilde\_gesamt\_II <- 2\*(dev\_tilde\_erst\_II+dev\_tilde\_zweit\_II)

> #dev\_tilde\_gesamt\_II

>

> leverage\_I <- fertige\_Daten\_w\_ik[,1] - dev\_tilde\_gesamt\_I

> leverage\_II <- fertige\_Daten\_w\_ik[,2] - dev\_tilde\_gesamt\_II

> #leverage\_I

> #leverage\_II

>

>

>

> #### Erzeugen leverage plot

> library(car)

> scatterplot(c(fertige\_Daten\_w\_ik[,"w\_ik\_neg"], fertige\_Daten\_w\_ik[,"w\_ik"]), c(leverage\_I, leverage\_II), main="Leverage plot for the random effects model", xlim=c(-3,3), ylim=c(0,4.5), xlab=expression('w'[ik]), ylab=expression('leverage'[ik]), regLine =F, smooth=F, boxplots=F )

> curve(-x^2+1, from=-3, to=3, col="red", lty="solid", add=T)

> curve(-x^2+2, from=-3, to=3, col="green", lty="dashed", add=T)

> curve(-x^2+3, from=-3, to=3, col="blueviolet", lty="dotted", add=T)

> curve(-x^2+4, from=-3, to=3, col="blue", lty="dotdash", add=T)

>

>

>

> # manuelle Berechnung von pD ----------------------------------------------

>

>

> #dev ist Std-Abweichung jedes einzelnen Werts

> # insg 22\*2 Wertepaare, also 44 Werte

> Hilf\_dev\_pD <- read.table("Hilf\_lePlo.txt", sep = "", header=F, skip=2, nrows=44)

> #Hilf\_dev\_pD

> Var\_manuell <- sum(Hilf\_dev\_pD[,2])^2/44

> pD\_manuell <- Var\_manuell/2

> pD\_manuell

[1] 19.2368

>

>

>

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