> ########## ########## Simulation Blocker Beispiel mit fixed 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/DietaryFat")

>

>

>

> #### 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("DietaryFat\_Data\_neu sortiert\_II\_ohneNA.csv", header=TRUE, sep = ";", quote = "\"", dec = ",", fill = TRUE, comment.char = "")

> #data

>

>

>

> #### Assignment data to stan

> NO =nrow(data)

> NT=max(data$Treatment\_t )

> NS=max(data$Studie)

> E=data$Explosure\_time\_E

> r=data$Erfolge\_r

> t=data$Treatment\_t

> s=data$Studie

> base=data$Treatment\_t

>

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

>

>

>

> #### Read in inits

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

+ list(d=c( NA, 0, 0),

+ sd=1,

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

+ }

>

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

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

+ sd=4,

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

+ }

>

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

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

+ sd=2,

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

+ }

>

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

> #all.inits

>

> #\* Hinweis:

> # Chain 3: Rejecting initial value:

> # Chain 3: Log probability evaluates to log(0), i.e. negative infinity.

> # Chain 3: Stan can't start sampling from this initial value.

> # => negative Werte zu positiv geändert

> # inits für Kette 1+2+4 auch. Aber augenscheinlich keine Probleme

>

>

>

> # Compiling

> m <- stan\_model('Model\_Fixed.stan')

recompiling to avoid crashing R session

> m <- stan\_model('Model\_Fixed.stan')

>

>

>

> # Simulation

> #stan\_samples <- sampling(m, data = data\_list, iter=20000, verbose=T, chain=4, control = list(adapt\_delta = 0.99)) # dauert zu lange

> stan\_samples <- sampling(m, data = data\_list, iter=30000, verbose=T, chain=4, control = list(adapt\_delta = 0.90)) # optimalere Lösung

CHECKING DATA AND PREPROCESSING FOR MODEL 'Model\_Fixed' NOW.

COMPILING MODEL 'Model\_Fixed' NOW.

STARTING SAMPLER FOR MODEL 'Model\_Fixed' NOW.

SAMPLING FOR MODEL 'Model\_Fixed' 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 / 30000 [ 0%] (Warmup)

Chain 1: Iteration: 3000 / 30000 [ 10%] (Warmup)

Chain 1: Iteration: 6000 / 30000 [ 20%] (Warmup)

Chain 1: Iteration: 9000 / 30000 [ 30%] (Warmup)

Chain 1: Iteration: 12000 / 30000 [ 40%] (Warmup)

Chain 1: Iteration: 15000 / 30000 [ 50%] (Warmup)

Chain 1: Iteration: 15001 / 30000 [ 50%] (Sampling)

Chain 1: Iteration: 18000 / 30000 [ 60%] (Sampling)

Chain 1: Iteration: 21000 / 30000 [ 70%] (Sampling)

Chain 1: Iteration: 24000 / 30000 [ 80%] (Sampling)

Chain 1: Iteration: 27000 / 30000 [ 90%] (Sampling)

Chain 1: Iteration: 30000 / 30000 [100%] (Sampling)

Chain 1:

Chain 1: Elapsed Time: 6.116 seconds (Warm-up)

Chain 1: 5.132 seconds (Sampling)

Chain 1: 11.248 seconds (Total)

Chain 1:

SAMPLING FOR MODEL 'Model\_Fixed' 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 / 30000 [ 0%] (Warmup)

Chain 2: Iteration: 3000 / 30000 [ 10%] (Warmup)

Chain 2: Iteration: 6000 / 30000 [ 20%] (Warmup)

Chain 2: Iteration: 9000 / 30000 [ 30%] (Warmup)

Chain 2: Iteration: 12000 / 30000 [ 40%] (Warmup)

Chain 2: Iteration: 15000 / 30000 [ 50%] (Warmup)

Chain 2: Iteration: 15001 / 30000 [ 50%] (Sampling)

Chain 2: Iteration: 18000 / 30000 [ 60%] (Sampling)

Chain 2: Iteration: 21000 / 30000 [ 70%] (Sampling)

Chain 2: Iteration: 24000 / 30000 [ 80%] (Sampling)

Chain 2: Iteration: 27000 / 30000 [ 90%] (Sampling)

Chain 2: Iteration: 30000 / 30000 [100%] (Sampling)

Chain 2:

Chain 2: Elapsed Time: 5.899 seconds (Warm-up)

Chain 2: 6.314 seconds (Sampling)

Chain 2: 12.213 seconds (Total)

Chain 2:

SAMPLING FOR MODEL 'Model\_Fixed' 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 / 30000 [ 0%] (Warmup)

Chain 3: Iteration: 3000 / 30000 [ 10%] (Warmup)

Chain 3: Iteration: 6000 / 30000 [ 20%] (Warmup)

Chain 3: Iteration: 9000 / 30000 [ 30%] (Warmup)

Chain 3: Iteration: 12000 / 30000 [ 40%] (Warmup)

Chain 3: Iteration: 15000 / 30000 [ 50%] (Warmup)

Chain 3: Iteration: 15001 / 30000 [ 50%] (Sampling)

Chain 3: Iteration: 18000 / 30000 [ 60%] (Sampling)

Chain 3: Iteration: 21000 / 30000 [ 70%] (Sampling)

Chain 3: Iteration: 24000 / 30000 [ 80%] (Sampling)

Chain 3: Iteration: 27000 / 30000 [ 90%] (Sampling)

Chain 3: Iteration: 30000 / 30000 [100%] (Sampling)

Chain 3:

Chain 3: Elapsed Time: 5.777 seconds (Warm-up)

Chain 3: 5.087 seconds (Sampling)

Chain 3: 10.864 seconds (Total)

Chain 3:

SAMPLING FOR MODEL 'Model\_Fixed' 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 / 30000 [ 0%] (Warmup)

Chain 4: Iteration: 3000 / 30000 [ 10%] (Warmup)

Chain 4: Iteration: 6000 / 30000 [ 20%] (Warmup)

Chain 4: Iteration: 9000 / 30000 [ 30%] (Warmup)

Chain 4: Iteration: 12000 / 30000 [ 40%] (Warmup)

Chain 4: Iteration: 15000 / 30000 [ 50%] (Warmup)

Chain 4: Iteration: 15001 / 30000 [ 50%] (Sampling)

Chain 4: Iteration: 18000 / 30000 [ 60%] (Sampling)

Chain 4: Iteration: 21000 / 30000 [ 70%] (Sampling)

Chain 4: Iteration: 24000 / 30000 [ 80%] (Sampling)

Chain 4: Iteration: 27000 / 30000 [ 90%] (Sampling)

Chain 4: Iteration: 30000 / 30000 [100%] (Sampling)

Chain 4:

Chain 4: Elapsed Time: 5.568 seconds (Warm-up)

Chain 4: 5.131 seconds (Sampling)

Chain 4: 10.699 seconds (Total)

Chain 4:

>

>

>

> #### Plotting and summarizing the posterior distribution

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

Inference for Stan model: Model\_Fixed.

4 chains, each with iter=30000; warmup=15000; thin=1;

post-warmup draws per chain=15000, total post-warmup draws=60000.

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

d[1] 0.00 0.00 0.32 -0.62 -0.21 0.00 0.21 0.61 128952 1

d[2] 0.00 0.00 0.30 -0.60 -0.21 0.00 0.21 0.60 120571 1

mu[1] -2.82 0.00 0.07 -2.95 -2.86 -2.82 -2.77 -2.69 125396 1

mu[2] -2.11 0.00 0.24 -2.59 -2.26 -2.10 -1.95 -1.67 114692 1

mu[3] -2.75 0.00 0.14 -3.03 -2.84 -2.75 -2.66 -2.49 113989 1

mu[4] -2.90 0.00 0.04 -2.99 -2.93 -2.90 -2.88 -2.82 124586 1

mu[5] -3.12 0.00 0.12 -3.36 -3.20 -3.12 -3.04 -2.89 119087 1

mu[6] -2.71 0.00 0.09 -2.90 -2.77 -2.71 -2.65 -2.54 122383 1

mu[7] -2.94 0.00 0.29 -3.55 -3.13 -2.93 -2.73 -2.39 114865 1

mu[8] -3.28 0.00 0.12 -3.52 -3.36 -3.28 -3.20 -3.06 121244 1

mu[9] -2.18 0.00 0.05 -2.28 -2.21 -2.18 -2.14 -2.08 123230 1

mu[10] -3.30 0.00 0.30 -3.92 -3.49 -3.29 -3.09 -2.74 116652 1

A[1] -3.00 0.00 0.56 -4.10 -3.38 -3.00 -2.62 -1.90 122909 1

A[2] -3.00 0.00 0.56 -4.10 -3.38 -3.00 -2.62 -1.90 120623 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 0.30 -0.60 -0.21 0.00 0.21 0.60 120571 1

lambda[1] 0.06 0.00 0.00 0.05 0.06 0.06 0.06 0.07 125836 1

lambda[2] 0.12 0.00 0.03 0.07 0.10 0.12 0.14 0.19 118767 1

lambda[3] 0.06 0.00 0.01 0.05 0.06 0.06 0.07 0.08 116240 1

lambda[4] 0.05 0.00 0.00 0.05 0.05 0.05 0.06 0.06 124147 1

lambda[5] 0.04 0.00 0.01 0.03 0.04 0.04 0.05 0.06 119946 1

lambda[6] 0.07 0.00 0.01 0.06 0.06 0.07 0.07 0.08 123984 1

lambda[7] 0.06 0.00 0.02 0.03 0.04 0.05 0.07 0.09 120209 1

lambda[8] 0.04 0.00 0.00 0.03 0.03 0.04 0.04 0.05 123027 1

lambda[9] 0.11 0.00 0.01 0.10 0.11 0.11 0.12 0.13 123926 1

lambda[10] 0.04 0.00 0.01 0.02 0.03 0.04 0.05 0.06 120814 1

lambda[11] 0.06 0.00 0.00 0.05 0.06 0.06 0.06 0.07 125836 1

lambda[12] 0.12 0.00 0.03 0.07 0.10 0.12 0.14 0.19 118767 1

lambda[13] 0.06 0.00 0.01 0.05 0.06 0.06 0.07 0.08 116240 1

lambda[14] 0.05 0.00 0.00 0.05 0.05 0.05 0.06 0.06 124147 1

lambda[15] 0.04 0.00 0.01 0.03 0.04 0.04 0.05 0.06 119946 1

lambda[16] 0.07 0.00 0.01 0.06 0.06 0.07 0.07 0.08 123984 1

lambda[17] 0.06 0.00 0.02 0.03 0.04 0.05 0.07 0.09 120209 1

lambda[18] 0.04 0.00 0.00 0.03 0.03 0.04 0.04 0.05 123027 1

lambda[19] 0.11 0.00 0.01 0.10 0.11 0.11 0.12 0.13 123926 1

lambda[20] 0.04 0.00 0.01 0.02 0.03 0.04 0.05 0.06 120814 1

lambda[21] 0.12 0.00 0.03 0.07 0.10 0.12 0.14 0.19 118767 1

theta[1] 114.56 0.02 7.53 100.22 109.40 114.39 119.53 129.77 125836 1

theta[2] 5.44 0.00 1.27 3.26 4.53 5.33 6.22 8.20 118767 1

theta[3] 25.38 0.01 3.51 18.94 22.94 25.21 27.65 32.70 116240 1

theta[4] 258.41 0.03 11.35 236.63 250.66 258.26 265.96 281.17 124147 1

theta[5] 31.84 0.01 3.87 24.72 29.16 31.68 34.34 39.84 119946 1

theta[6] 58.86 0.02 5.37 48.71 55.19 58.73 62.35 69.92 123984 1

theta[7] 4.85 0.00 1.41 2.53 3.84 4.71 5.71 8.02 120209 1

theta[8] 38.14 0.01 4.40 30.04 35.09 37.98 41.01 47.29 123027 1

theta[9] 175.19 0.03 9.19 157.74 168.90 175.03 181.32 193.59 123926 1

theta[10] 4.84 0.00 1.43 2.49 3.81 4.68 5.69 8.05 120814 1

theta[11] 115.04 0.02 7.56 100.64 109.85 114.86 120.03 130.31 125836 1

theta[12] 5.15 0.00 1.20 3.09 4.29 5.05 5.90 7.77 118767 1

theta[13] 24.11 0.01 3.34 17.99 21.79 23.96 26.27 31.07 116240 1

theta[14] 264.33 0.03 11.61 242.05 256.40 264.17 272.05 287.61 124147 1

theta[15] 33.44 0.01 4.06 25.97 30.62 33.28 36.07 41.85 119946 1

theta[16] 59.53 0.02 5.43 49.26 55.82 59.39 63.06 70.71 123984 1

theta[17] 5.03 0.00 1.46 2.62 3.98 4.88 5.92 8.31 120209 1

theta[18] 35.43 0.01 4.09 27.90 32.59 35.27 38.09 43.92 123027 1

theta[19] 180.18 0.03 9.45 162.23 173.71 180.02 186.49 199.10 123926 1

theta[20] 4.76 0.00 1.41 2.45 3.75 4.60 5.60 7.92 120814 1

theta[21] 4.74 0.00 1.11 2.84 3.95 4.65 5.43 7.15 118767 1

dev[1] 0.51 0.00 0.72 0.00 0.05 0.23 0.67 2.55 23688 1

dev[2] 5.55 0.01 2.08 2.16 4.04 5.31 6.79 10.19 116148 1

dev[3] 0.54 0.00 0.75 0.00 0.05 0.25 0.72 2.69 31350 1

dev[4] 0.90 0.01 1.14 0.00 0.11 0.48 1.27 4.08 47358 1

dev[5] 0.48 0.00 0.68 0.00 0.05 0.22 0.63 2.42 25836 1

dev[6] 1.16 0.01 1.37 0.00 0.18 0.68 1.65 4.93 53518 1

dev[7] 1.08 0.00 1.15 0.00 0.22 0.72 1.56 4.14 77086 1

dev[8] 3.35 0.01 2.41 0.15 1.54 2.89 4.65 9.23 100906 1

dev[9] 0.51 0.00 0.71 0.00 0.05 0.23 0.67 2.52 25618 1

dev[10] 2.32 0.01 1.71 0.11 1.04 1.96 3.20 6.53 101674 1

dev[11] 0.62 0.00 0.86 0.00 0.07 0.29 0.84 3.06 31877 1

dev[12] 0.28 0.00 0.40 0.00 0.03 0.13 0.37 1.41 24040 1

dev[13] 1.13 0.00 1.26 0.00 0.20 0.71 1.64 4.52 68049 1

dev[14] 0.60 0.00 0.84 0.00 0.06 0.27 0.80 3.02 29623 1

dev[15] 1.35 0.01 1.45 0.00 0.27 0.89 1.96 5.19 68971 1

dev[16] 2.79 0.01 2.17 0.06 1.15 2.34 3.92 8.23 94378 1

dev[17] 4.92 0.01 2.35 1.31 3.19 4.59 6.28 10.39 115588 1

dev[18] 0.87 0.01 1.10 0.00 0.11 0.46 1.22 3.93 45673 1

dev[19] 0.69 0.00 0.93 0.00 0.08 0.33 0.95 3.33 39210 1

dev[20] 4.49 0.01 2.23 1.11 2.86 4.15 5.76 9.71 114925 1

dev[21] 0.90 0.00 0.85 0.01 0.25 0.67 1.30 3.08 83699 1

T[1] 0.06 0.00 0.04 0.02 0.03 0.05 0.07 0.15 77924 1

T[2] 0.06 0.00 0.04 0.01 0.03 0.05 0.08 0.18 68985 1

totalresdev 35.03 0.03 7.58 22.33 29.60 34.31 39.65 51.88 59934 1

lp\_\_ 5275.53 0.02 2.67 5269.36 5273.96 5275.87 5277.47 5279.73 22624 1

Samples were drawn using NUTS(diag\_e) at Sun Aug 25 16:44:27 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)

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

> Stan\_summary

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

d[2] 1.723518e-04 0.0008773373 0.30464075 -5.981661e-01 0.5969997 120570.99 0.9999590

d\_II[2] 1.723518e-04 0.0008773373 0.30464075 -5.981661e-01 0.5969997 120570.99 0.9999590

T[1] 5.833227e-02 0.0001284313 0.03585150 1.653409e-02 0.1501890 77924.33 0.9999999

T[2] 6.123277e-02 0.0001660729 0.04361892 1.425648e-02 0.1750104 68984.59 1.0000805

totalresdev 3.502897e+01 0.0309520575 7.57752293 2.233471e+01 51.8788899 59934.30 0.9999618

dev[1] 5.097018e-01 0.0046725821 0.71914588 5.166763e-04 2.5536752 23687.55 1.0001595

dev[2] 5.547209e+00 0.0060963392 2.07765901 2.159233e+00 10.1894900 116147.61 0.9999752

dev[3] 5.377878e-01 0.0042420376 0.75109517 5.223329e-04 2.6898513 31350.24 1.0000528

dev[4] 9.039156e-01 0.0052487975 1.14223898 1.040024e-03 4.0766682 47358.11 1.0000104

dev[5] 4.803446e-01 0.0042413118 0.68173216 4.403487e-04 2.4233746 25836.11 0.9999886

dev[6] 1.161078e+00 0.0059235738 1.37035861 2.039289e-03 4.9278925 53518.12 1.0000363

dev[7] 1.075747e+00 0.0041398851 1.14941142 2.432176e-03 4.1394297 77085.81 0.9999683

dev[8] 3.347464e+00 0.0075717532 2.40522606 1.495086e-01 9.2332013 100906.45 0.9999684

dev[9] 5.060276e-01 0.0044192805 0.70733193 4.707102e-04 2.5165209 25617.89 0.9999978

dev[10] 2.317011e+00 0.0053510139 1.70624447 1.077307e-01 6.5347890 101674.11 0.9999580

dev[11] 6.231077e-01 0.0048134176 0.85939141 6.803299e-04 3.0579790 31876.82 1.0001133

dev[12] 2.778454e-01 0.0025621719 0.39726075 2.470655e-04 1.4129894 24040.02 1.0001051

dev[13] 1.130307e+00 0.0048269635 1.25917098 1.865882e-03 4.5198671 68048.94 1.0000035

dev[14] 6.011919e-01 0.0048716003 0.83846629 5.983936e-04 3.0225415 29622.92 1.0000667

dev[15] 1.351507e+00 0.0055099916 1.44705473 3.301244e-03 5.1941013 68971.24 0.9999500

dev[16] 2.792606e+00 0.0070669349 2.17103725 5.671320e-02 8.2329191 94378.35 0.9999591

dev[17] 4.916046e+00 0.0069195618 2.35252670 1.313126e+00 10.3882349 115587.78 0.9999543

dev[18] 8.692216e-01 0.0051349922 1.09740762 1.063631e-03 3.9261984 45672.67 0.9999631

dev[19] 6.937907e-01 0.0047174929 0.93413330 7.733661e-04 3.3313314 39209.85 1.0000033

dev[20] 4.486674e+00 0.0065862387 2.23277103 1.109134e+00 9.7089070 114924.72 0.9999541

dev[21] 9.003908e-01 0.0029486094 0.85305463 5.117729e-03 3.0839817 83698.80 0.9999810

theta[1] 1.145590e+02 0.0212214111 7.52794035 1.002203e+02 129.7730599 125835.69 0.9999888

theta[2] 5.440072e+00 0.0036863574 1.27041498 3.261938e+00 8.1987235 118767.15 0.9999758

theta[3] 2.537842e+01 0.0103041497 3.51310269 1.893639e+01 32.6991238 116240.47 0.9999648

theta[4] 2.584117e+02 0.0322206973 11.35282071 2.366307e+02 281.1669183 124147.42 0.9999574

theta[5] 3.183839e+01 0.0111624171 3.86590018 2.472227e+01 39.8435718 119945.73 0.9999489

theta[6] 5.886231e+01 0.0152481092 5.36906000 4.870624e+01 69.9229836 123983.68 0.9999490

theta[7] 4.854329e+00 0.0040721962 1.41187603 2.527458e+00 8.0195099 120208.66 0.9999527

theta[8] 3.814242e+01 0.0125527542 4.40290003 3.004175e+01 47.2930455 123026.76 0.9999684

theta[9] 1.751906e+02 0.0261108370 9.19184220 1.577370e+02 193.5880114 123926.31 0.9999787

theta[10] 4.836987e+00 0.0041169614 1.43098872 2.490962e+00 8.0532521 120814.44 0.9999525

theta[11] 1.150371e+02 0.0213099721 7.55935586 1.006385e+02 130.3146272 125835.69 0.9999888

theta[12] 5.153095e+00 0.0034918936 1.20339768 3.089864e+00 7.7662220 118767.15 0.9999758

theta[13] 2.411434e+01 0.0097909061 3.33811715 1.799318e+01 31.0704000 116240.47 0.9999648

theta[14] 2.643308e+02 0.0329587323 11.61286412 2.420509e+02 287.6072210 124147.42 0.9999574

theta[15] 3.344144e+01 0.0117244409 4.06054690 2.596703e+01 41.8496817 119945.73 0.9999489

theta[16] 5.952742e+01 0.0154204042 5.42972734 4.925660e+01 70.7130738 123983.68 0.9999490

theta[17] 5.031252e+00 0.0042206133 1.46333393 2.619575e+00 8.3117928 120208.66 0.9999527

theta[18] 3.542605e+01 0.0116587895 4.08934038 2.790228e+01 43.9249948 123026.76 0.9999684

theta[19] 1.801831e+02 0.0268549282 9.45378589 1.622321e+02 199.1047682 123926.31 0.9999787

theta[20] 4.759595e+00 0.0040510900 1.40809291 2.451107e+00 7.9244001 120814.44 0.9999525

theta[21] 4.741347e+00 0.0032128803 1.10724242 2.842974e+00 7.1456764 118767.15 0.9999758

>

>

>

> #### Additional Lines for Median

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

> Median\_d2

[1] 2.975896e-05

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

> Median\_T1

[1] 0.04984458

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

> Median\_T2

[1] 0.04999752

>

>

>

>

>

>

> # 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. : 0.0058 Min. : 0 Min. : 1.0 Min. :0.00000 Min. :-5280

1st Qu.:0.86 1st Qu.: 0.5356 1st Qu.: 3 1st Qu.: 7.0 1st Qu.:0.00000 1st Qu.:-5271

Median :0.94 Median : 0.5551 Median : 3 Median : 7.0 Median :0.00000 Median :-5269

Mean :0.91 Mean : 0.5665 Mean : 3 Mean : 7.4 Mean :0.00038 Mean :-5267

3rd Qu.:0.99 3rd Qu.: 0.5944 3rd Qu.: 3 3rd Qu.: 7.0 3rd Qu.:0.00000 3rd Qu.:-5266

Max. :1.00 Max. :11.9940 Max. :10 Max. :1023.0 Max. :1.00000 Max. :38179

> # each chain separately

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

[[1]]

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

Min. :0.00 Min. : 0.0058 Min. : 0 Min. : 1.0 Min. :0.00000 Min. :-5280

1st Qu.:0.86 1st Qu.: 0.5356 1st Qu.: 3 1st Qu.: 7.0 1st Qu.:0.00000 1st Qu.:-5271

Median :0.94 Median : 0.5356 Median : 3 Median : 7.0 Median :0.00000 Median :-5269

Mean :0.91 Mean : 0.5666 Mean : 3 Mean : 7.5 Mean :0.00043 Mean :-5268

3rd Qu.:0.99 3rd Qu.: 0.5935 3rd Qu.: 3 3rd Qu.: 7.0 3rd Qu.:0.00000 3rd Qu.:-5266

Max. :1.00 Max. :11.9940 Max. :10 Max. :1023.0 Max. :1.00000 Max. : 7000

[[2]]

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

Min. :0.00 Min. : 0.0058 Min. :0 Min. : 1.0 Min. :0e+00 Min. :-5279

1st Qu.:0.86 1st Qu.: 0.5378 1st Qu.:3 1st Qu.: 7.0 1st Qu.:0e+00 1st Qu.:-5271

Median :0.94 Median : 0.5378 Median :3 Median : 7.0 Median :0e+00 Median :-5269

Mean :0.91 Mean : 0.5678 Mean :3 Mean : 7.4 Mean :4e-04 Mean :-5267

3rd Qu.:0.99 3rd Qu.: 0.5947 3rd Qu.:3 3rd Qu.: 7.0 3rd Qu.:0e+00 3rd Qu.:-5266

Max. :1.00 Max. :11.9940 Max. :8 Max. :419.0 Max. :1e+00 Max. :13527

[[3]]

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

Min. :0.00 Min. : 0.0058 Min. :0 Min. : 1.0 Min. :0.00000 Min. :-5279

1st Qu.:0.87 1st Qu.: 0.5015 1st Qu.:3 1st Qu.: 7.0 1st Qu.:0.00000 1st Qu.:-5271

Median :0.95 Median : 0.5015 Median :3 Median : 7.0 Median :0.00000 Median :-5269

Mean :0.91 Mean : 0.5491 Mean :3 Mean : 7.4 Mean :0.00033 Mean :-5268

3rd Qu.:0.99 3rd Qu.: 0.5943 3rd Qu.:3 3rd Qu.: 7.0 3rd Qu.:0.00000 3rd Qu.:-5266

Max. :1.00 Max. :11.9940 Max. :9 Max. :539.0 Max. :1.00000 Max. : 5329

[[4]]

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

Min. :0.00 Min. : 0.0058 Min. :0 Min. : 1.0 Min. :0.00000 Min. :-5279

1st Qu.:0.85 1st Qu.: 0.5666 1st Qu.:3 1st Qu.: 7.0 1st Qu.:0.00000 1st Qu.:-5271

Median :0.94 Median : 0.5666 Median :3 Median : 7.0 Median :0.00000 Median :-5269

Mean :0.90 Mean : 0.5826 Mean :3 Mean : 7.4 Mean :0.00033 Mean :-5266

3rd Qu.:0.99 3rd Qu.: 0.5954 3rd Qu.:3 3rd Qu.: 7.0 3rd Qu.:0.00000 3rd Qu.:-5266

Max. :1.00 Max. :11.9940 Max. :8 Max. :351.0 Max. :1.00000 Max. :38179

> 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\_\_ 5275.533 0.01776579 2.672219 5269.359 5279.73 22624.3 1.000056

> # weitere Möglichkeit: Package 'shinystan'

>

>

>

>

>

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

>

>

>

>

> 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]"))$summary

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

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

> Hilf\_dev\_pD <- read.table("Hilf\_pD.txt", sep = "", header=F, skip=2, nrows=21)

> #dev ist Std-Abweichung jedes einzelnen Werts

> # insg 21 Werte

> Var\_manuell <- sum(Hilf\_dev\_pD[,2])^2/21

> Var\_manuell

[1] 114.9369

> pD\_manuell <- Var\_manuell/2

> pD\_manuell

[1] 57.46845

>

>

>

>

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