> ########## ########## Simulation Dietary Fat Beispiel mit Fixed Effects mit STAN ########## ##########

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

>

>

>

>

> ##### Clear data

> rm(list=ls())

>

>

>

> ##### Setting working directory

> setwd("C:/Users/IvanB/Desktop/Masterarbeit/Statistische Programme und Gibbs Sampler/STAN/Nachrechnen TSD2/Ex3 Diabetes")

>

>

>

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

> head(data) # Shows the first six entries

time Treatment\_t Erfolge\_r Gesamtanzahl\_n Studienarm Studie Studie\_test X.Study

1 5.8 1 43 1081 3 1 1 #MRC-E 38

2 4.7 1 29 416 2 2 2 #EWPH 32

3 3.0 1 140 1631 2 3 3 #SHEP 42

4 3.8 1 75 3272 2 4 4 #HAPPHY 33

5 4.0 1 302 6766 3 5 5 #ALLHAT 26

6 3.0 1 176 2511 2 6 6 #INSIGHT 35

> #data2 = read.table("Diabetes\_Data\_Rest.txt")

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

>

>

>

> ##### Assignment data to stan

> NO =nrow(data)

> #NO =48

> NT=max(data$Treatment\_t )

> NS=max(data$Studie)

> time=data$time

> r=data$Erfolge\_r

> t=data$Treatment\_t

> s=data$Studie

> base=data$Treatment\_t

> n=data$Gesamtanzahl\_n

>

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

>

>

>

> ##### Read in inits

> # Anmerkung: da cloglog als Link: wurden die Inits von STAN 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),

+

+

+ delta= c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ))

> # die anderen Inits müssen nach der Art von Inits1 angepasst werden

>

> # delta= 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)) # structure(.Data= , .Dim=c(22, 3))

> # 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=c(0, 0, 0, 0, 0, 0) ,

>

> 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= 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)) # structure(.Data= , .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= 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)) # structure(.Data= , .Dim=c(22, 3))

>

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

> #function(chain\_id = 1) {

>

>

>

> # Compiling

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

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

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

>

>

>

> stan\_samples <- sampling(m, data = data\_list, iter=30000, verbose=T, chain=3, init\_r=0.1)

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: 135.198 seconds (Warm-up)

Chain 1: 18.031 seconds (Sampling)

Chain 1: 153.229 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: 128.835 seconds (Warm-up)

Chain 2: 15.899 seconds (Sampling)

Chain 2: 144.734 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: 132.821 seconds (Warm-up)

Chain 3: 23.526 seconds (Sampling)

Chain 3: 156.347 seconds (Total)

Chain 3:

Warning messages:

1: There were 282 transitions after warmup that exceeded the maximum treedepth. Increase max\_treedepth above 10. See

http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

2: Examine the pairs() plot to diagnose sampling problems

3: The largest R-hat is 1.67, indicating chains have not mixed.

Running the chains for more iterations may help. See

http://mc-stan.org/misc/warnings.html#r-hat

4: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be unreliable.

Running the chains for more iterations may help. See

http://mc-stan.org/misc/warnings.html#bulk-ess

5: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quantiles may be unreliable.

Running the chains for more iterations may help. See

http://mc-stan.org/misc/warnings.html#tail-ess

> #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, i

>

>

>

>

>

> # Ausgabe posteriore Werte, Berechnung Median --------------------------

>

>

>

>

> ##### Plotting and summarizing the posterior distribution

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

Inference for Stan model: Model\_Fixed.

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

post-warmup draws per chain=15000, total post-warmup draws=45000.

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

d[1] 2.112147e+31 2.253807e+31 3.382323e+31 -6.149726e+29 5.708219e+29 1.427237e+30 3.968028e+31 9.546077e+31 2 2.34

d[2] 0.000000e+00 0.000000e+00 9.900000e-01 -1.940000e+00 -6.700000e-01 0.000000e+00 6.700000e-01 1.930000e+00 104628 1.00

d[3] 0.000000e+00 0.000000e+00 1.000000e+00 -1.940000e+00 -6.700000e-01 1.000000e-02 6.800000e-01 1.960000e+00 113230 1.00

d[4] 0.000000e+00 0.000000e+00 1.000000e+00 -1.950000e+00 -6.800000e-01 0.000000e+00 6.800000e-01 1.940000e+00 109120 1.00

d[5] 0.000000e+00 0.000000e+00 1.010000e+00 -1.950000e+00 -6.800000e-01 0.000000e+00 6.800000e-01 1.970000e+00 104986 1.00

d[6] 0.000000e+00 0.000000e+00 1.010000e+00 -1.970000e+00 -6.800000e-01 0.000000e+00 6.800000e-01 1.990000e+00 104736 1.00

mu[1] -5.270000e+00 0.000000e+00 9.000000e-02 -5.440000e+00 -5.330000e+00 -5.270000e+00 -5.210000e+00 -5.100000e+00 104102 1.00

mu[2] -4.210000e+00 0.000000e+00 1.300000e-01 -4.480000e+00 -4.300000e+00 -4.210000e+00 -4.120000e+00 -3.960000e+00 103085 1.00

mu[3] -3.550000e+00 0.000000e+00 6.000000e-02 -3.670000e+00 -3.590000e+00 -3.550000e+00 -3.510000e+00 -3.430000e+00 105325 1.00

mu[4] -4.970000e+00 0.000000e+00 8.000000e-02 -5.130000e+00 -5.030000e+00 -4.970000e+00 -4.920000e+00 -4.830000e+00 110162 1.00

mu[5] -4.590000e+00 0.000000e+00 4.000000e-02 -4.670000e+00 -4.620000e+00 -4.590000e+00 -4.560000e+00 -4.510000e+00 104412 1.00

mu[6] -3.820000e+00 0.000000e+00 6.000000e-02 -3.930000e+00 -3.860000e+00 -3.820000e+00 -3.780000e+00 -3.710000e+00 102911 1.00

mu[7] -4.170000e+00 0.000000e+00 5.000000e-02 -4.280000e+00 -4.210000e+00 -4.170000e+00 -4.130000e+00 -4.060000e+00 102721 1.00

mu[8] -3.250000e+00 0.000000e+00 2.400000e-01 -3.750000e+00 -3.410000e+00 -3.240000e+00 -3.080000e+00 -2.800000e+00 98631 1.00

mu[9] -4.530000e+00 0.000000e+00 5.000000e-02 -4.640000e+00 -4.570000e+00 -4.530000e+00 -4.490000e+00 -4.430000e+00 97306 1.00

mu[10] -2.720000e+00 0.000000e+00 3.000000e-02 -2.790000e+00 -2.740000e+00 -2.720000e+00 -2.700000e+00 -2.660000e+00 104869 1.00

mu[11] -4.550000e+00 0.000000e+00 6.000000e-02 -4.670000e+00 -4.590000e+00 -4.550000e+00 -4.510000e+00 -4.430000e+00 100710 1.00

mu[12] -3.740000e+00 0.000000e+00 4.000000e-02 -3.820000e+00 -3.770000e+00 -3.740000e+00 -3.720000e+00 -3.670000e+00 105797 1.00

mu[13] -3.780000e+00 0.000000e+00 5.000000e-02 -3.880000e+00 -3.810000e+00 -3.780000e+00 -3.740000e+00 -3.680000e+00 107833 1.00

mu[14] -4.280000e+00 0.000000e+00 7.000000e-02 -4.420000e+00 -4.330000e+00 -4.280000e+00 -4.240000e+00 -4.150000e+00 110184 1.00

mu[15] -3.130000e+00 0.000000e+00 8.000000e-02 -3.290000e+00 -3.190000e+00 -3.130000e+00 -3.080000e+00 -2.980000e+00 107546 1.00

mu[16] -4.350000e+00 0.000000e+00 6.000000e-02 -4.460000e+00 -4.390000e+00 -4.350000e+00 -4.310000e+00 -4.240000e+00 109791 1.00

mu[17] -3.980000e+00 0.000000e+00 3.000000e-02 -4.040000e+00 -4.000000e+00 -3.980000e+00 -3.970000e+00 -3.930000e+00 110364 1.00

mu[18] -4.540000e+00 0.000000e+00 5.000000e-02 -4.630000e+00 -4.570000e+00 -4.540000e+00 -4.510000e+00 -4.450000e+00 104679 1.00

mu[19] -3.910000e+00 0.000000e+00 3.000000e-02 -3.970000e+00 -3.930000e+00 -3.910000e+00 -3.900000e+00 -3.860000e+00 103126 1.00

mu[20] -4.440000e+00 0.000000e+00 4.000000e-02 -4.510000e+00 -4.460000e+00 -4.440000e+00 -4.410000e+00 -4.370000e+00 100852 1.00

mu[21] -4.170000e+00 0.000000e+00 4.000000e-02 -4.260000e+00 -4.200000e+00 -4.170000e+00 -4.150000e+00 -4.090000e+00 105639 1.00

mu[22] -3.240000e+00 0.000000e+00 3.000000e-02 -3.290000e+00 -3.260000e+00 -3.240000e+00 -3.220000e+00 -3.190000e+00 108523 1.00

A[1] -4.200000e+00 0.000000e+00 9.500000e-01 -6.060000e+00 -4.850000e+00 -4.200000e+00 -3.560000e+00 -2.350000e+00 104576 1.00

A[2] -4.200000e+00 0.000000e+00 9.500000e-01 -6.040000e+00 -4.850000e+00 -4.190000e+00 -3.560000e+00 -2.360000e+00 102834 1.00

A[3] -4.200000e+00 0.000000e+00 9.600000e-01 -6.080000e+00 -4.840000e+00 -4.200000e+00 -3.550000e+00 -2.330000e+00 104204 1.00

A[4] -4.200000e+00 0.000000e+00 9.400000e-01 -6.050000e+00 -4.840000e+00 -4.210000e+00 -3.570000e+00 -2.340000e+00 112182 1.00

A[5] -4.200000e+00 0.000000e+00 9.500000e-01 -6.070000e+00 -4.840000e+00 -4.200000e+00 -3.560000e+00 -2.340000e+00 110545 1.00

A[6] -4.200000e+00 0.000000e+00 9.400000e-01 -6.040000e+00 -4.840000e+00 -4.200000e+00 -3.570000e+00 -2.340000e+00 107047 1.00

d\_II[1] 0.000000e+00 NaN 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 NaN NaN

d\_II[2] 0.000000e+00 0.000000e+00 9.900000e-01 -1.940000e+00 -6.700000e-01 0.000000e+00 6.700000e-01 1.930000e+00 104628 1.00

d\_II[3] 0.000000e+00 0.000000e+00 1.000000e+00 -1.940000e+00 -6.700000e-01 1.000000e-02 6.800000e-01 1.960000e+00 113230 1.00

d\_II[4] 0.000000e+00 0.000000e+00 1.000000e+00 -1.950000e+00 -6.800000e-01 0.000000e+00 6.800000e-01 1.940000e+00 109120 1.00

d\_II[5] 0.000000e+00 0.000000e+00 1.010000e+00 -1.950000e+00 -6.800000e-01 0.000000e+00 6.800000e-01 1.970000e+00 104986 1.00

d\_II[6] 0.000000e+00 0.000000e+00 1.010000e+00 -1.970000e+00 -6.800000e-01 0.000000e+00 6.800000e-01 1.990000e+00 104736 1.00

p[1] -3.510000e+00 0.000000e+00 9.000000e-02 -3.690000e+00 -3.570000e+00 -3.510000e+00 -3.450000e+00 -3.350000e+00 104102 1.00

p[2] -2.660000e+00 0.000000e+00 1.300000e-01 -2.930000e+00 -2.750000e+00 -2.660000e+00 -2.570000e+00 -2.410000e+00 103085 1.00

p[3] -2.450000e+00 0.000000e+00 6.000000e-02 -2.570000e+00 -2.490000e+00 -2.450000e+00 -2.410000e+00 -2.340000e+00 105325 1.00

p[4] -3.640000e+00 0.000000e+00 8.000000e-02 -3.790000e+00 -3.690000e+00 -3.640000e+00 -3.590000e+00 -3.490000e+00 110162 1.00

p[5] -3.210000e+00 0.000000e+00 4.000000e-02 -3.290000e+00 -3.230000e+00 -3.210000e+00 -3.180000e+00 -3.130000e+00 104412 1.00

p[6] -2.720000e+00 0.000000e+00 6.000000e-02 -2.830000e+00 -2.760000e+00 -2.720000e+00 -2.680000e+00 -2.610000e+00 102911 1.00

p[7] -2.760000e+00 0.000000e+00 5.000000e-02 -2.870000e+00 -2.790000e+00 -2.760000e+00 -2.720000e+00 -2.650000e+00 102721 1.00

p[8] -3.250000e+00 0.000000e+00 2.400000e-01 -3.750000e+00 -3.410000e+00 -3.240000e+00 -3.080000e+00 -2.800000e+00 98631 1.00

p[9] -3.340000e+00 0.000000e+00 5.000000e-02 -3.440000e+00 -3.370000e+00 -3.340000e+00 -3.300000e+00 -3.230000e+00 97306 1.00

p[10] -1.620000e+00 0.000000e+00 3.000000e-02 -1.690000e+00 -1.650000e+00 -1.620000e+00 -1.600000e+00 -1.560000e+00 104869 1.00

p[11] -3.050000e+00 0.000000e+00 6.000000e-02 -3.170000e+00 -3.090000e+00 -3.050000e+00 -3.010000e+00 -2.930000e+00 100710 1.00

p[12] -2.180000e+00 0.000000e+00 4.000000e-02 -2.250000e+00 -2.200000e+00 -2.180000e+00 -2.150000e+00 -2.100000e+00 105797 1.00

p[13] -2.650000e+00 0.000000e+00 5.000000e-02 -2.750000e+00 -2.680000e+00 -2.650000e+00 -2.610000e+00 -2.550000e+00 107833 1.00

p[14] -2.980000e+00 0.000000e+00 7.000000e-02 -3.110000e+00 -3.020000e+00 -2.980000e+00 -2.930000e+00 -2.850000e+00 110184 1.00

p[15] -1.800000e+00 0.000000e+00 8.000000e-02 -1.960000e+00 -1.850000e+00 -1.800000e+00 -1.740000e+00 -1.640000e+00 107546 1.00

p[16] -2.960000e+00 0.000000e+00 6.000000e-02 -3.080000e+00 -3.000000e+00 -2.960000e+00 -2.920000e+00 -2.850000e+00 109791 1.00

p[17] -2.280000e+00 0.000000e+00 3.000000e-02 -2.330000e+00 -2.300000e+00 -2.280000e+00 -2.260000e+00 -2.230000e+00 110364 1.00

p[18] -3.040000e+00 0.000000e+00 5.000000e-02 -3.130000e+00 -3.070000e+00 -3.040000e+00 -3.010000e+00 -2.950000e+00 104679 1.00

p[19] -2.530000e+00 0.000000e+00 3.000000e-02 -2.580000e+00 -2.550000e+00 -2.530000e+00 -2.510000e+00 -2.470000e+00 103126 1.00

p[20] -2.630000e+00 0.000000e+00 4.000000e-02 -2.700000e+00 -2.650000e+00 -2.630000e+00 -2.600000e+00 -2.560000e+00 100852 1.00

p[21] -2.610000e+00 0.000000e+00 4.000000e-02 -2.690000e+00 -2.630000e+00 -2.610000e+00 -2.580000e+00 -2.520000e+00 105639 1.00

p[22] -1.810000e+00 0.000000e+00 3.000000e-02 -1.860000e+00 -1.820000e+00 -1.810000e+00 -1.790000e+00 -1.760000e+00 108523 1.00

p[23] -3.510000e+00 0.000000e+00 9.000000e-02 -3.690000e+00 -3.570000e+00 -3.510000e+00 -3.450000e+00 -3.350000e+00 104102 1.00

p[24] -2.660000e+00 0.000000e+00 1.300000e-01 -2.930000e+00 -2.750000e+00 -2.660000e+00 -2.570000e+00 -2.410000e+00 103085 1.00

p[25] -2.450000e+00 0.000000e+00 6.000000e-02 -2.570000e+00 -2.490000e+00 -2.450000e+00 -2.410000e+00 -2.340000e+00 105325 1.00

p[26] -3.640000e+00 0.000000e+00 8.000000e-02 -3.790000e+00 -3.690000e+00 -3.640000e+00 -3.590000e+00 -3.490000e+00 110162 1.00

p[27] -3.210000e+00 0.000000e+00 4.000000e-02 -3.290000e+00 -3.230000e+00 -3.210000e+00 -3.180000e+00 -3.130000e+00 104412 1.00

p[28] -2.720000e+00 0.000000e+00 6.000000e-02 -2.830000e+00 -2.760000e+00 -2.720000e+00 -2.680000e+00 -2.610000e+00 102911 1.00

p[29] -2.760000e+00 0.000000e+00 5.000000e-02 -2.870000e+00 -2.790000e+00 -2.760000e+00 -2.720000e+00 -2.650000e+00 102721 1.00

p[30] -3.250000e+00 0.000000e+00 2.400000e-01 -3.750000e+00 -3.410000e+00 -3.240000e+00 -3.080000e+00 -2.800000e+00 98631 1.00

p[31] -3.340000e+00 0.000000e+00 5.000000e-02 -3.440000e+00 -3.370000e+00 -3.340000e+00 -3.300000e+00 -3.230000e+00 97306 1.00

p[32] -1.620000e+00 0.000000e+00 3.000000e-02 -1.690000e+00 -1.650000e+00 -1.620000e+00 -1.600000e+00 -1.560000e+00 104869 1.00

p[33] -3.050000e+00 0.000000e+00 6.000000e-02 -3.170000e+00 -3.090000e+00 -3.050000e+00 -3.010000e+00 -2.930000e+00 100710 1.00

p[34] -2.180000e+00 0.000000e+00 4.000000e-02 -2.250000e+00 -2.200000e+00 -2.180000e+00 -2.150000e+00 -2.100000e+00 105797 1.00

p[35] -2.650000e+00 0.000000e+00 5.000000e-02 -2.750000e+00 -2.680000e+00 -2.650000e+00 -2.610000e+00 -2.550000e+00 107833 1.00

p[36] -2.980000e+00 0.000000e+00 7.000000e-02 -3.110000e+00 -3.020000e+00 -2.980000e+00 -2.930000e+00 -2.850000e+00 110184 1.00

p[37] -1.800000e+00 0.000000e+00 8.000000e-02 -1.960000e+00 -1.850000e+00 -1.800000e+00 -1.740000e+00 -1.640000e+00 107546 1.00

p[38] -2.960000e+00 0.000000e+00 6.000000e-02 -3.080000e+00 -3.000000e+00 -2.960000e+00 -2.920000e+00 -2.850000e+00 109791 1.00

p[39] -2.280000e+00 0.000000e+00 3.000000e-02 -2.330000e+00 -2.300000e+00 -2.280000e+00 -2.260000e+00 -2.230000e+00 110364 1.00

p[40] -3.040000e+00 0.000000e+00 5.000000e-02 -3.130000e+00 -3.070000e+00 -3.040000e+00 -3.010000e+00 -2.950000e+00 104679 1.00

p[41] -2.530000e+00 0.000000e+00 3.000000e-02 -2.580000e+00 -2.550000e+00 -2.530000e+00 -2.510000e+00 -2.470000e+00 103126 1.00

p[42] -2.630000e+00 0.000000e+00 4.000000e-02 -2.700000e+00 -2.650000e+00 -2.630000e+00 -2.600000e+00 -2.560000e+00 100852 1.00

p[43] -2.610000e+00 0.000000e+00 4.000000e-02 -2.690000e+00 -2.630000e+00 -2.610000e+00 -2.580000e+00 -2.520000e+00 105639 1.00

p[44] -1.810000e+00 0.000000e+00 3.000000e-02 -1.860000e+00 -1.820000e+00 -1.810000e+00 -1.790000e+00 -1.760000e+00 108523 1.00

p[45] -3.510000e+00 0.000000e+00 9.000000e-02 -3.690000e+00 -3.570000e+00 -3.510000e+00 -3.450000e+00 -3.350000e+00 104102 1.00

p[46] -3.210000e+00 0.000000e+00 4.000000e-02 -3.290000e+00 -3.230000e+00 -3.210000e+00 -3.180000e+00 -3.130000e+00 104412 1.00

p[47] -1.800000e+00 0.000000e+00 8.000000e-02 -1.960000e+00 -1.850000e+00 -1.800000e+00 -1.740000e+00 -1.640000e+00 107546 1.00

p[48] -2.960000e+00 0.000000e+00 6.000000e-02 -3.080000e+00 -3.000000e+00 -2.960000e+00 -2.920000e+00 -2.850000e+00 109791 1.00

p\_cloglog[1] 3.000000e-02 0.000000e+00 0.000000e+00 2.000000e-02 3.000000e-02 3.000000e-02 3.000000e-02 3.000000e-02 104123 1.00

p\_cloglog[2] 7.000000e-02 0.000000e+00 1.000000e-02 5.000000e-02 6.000000e-02 7.000000e-02 7.000000e-02 9.000000e-02 105527 1.00

p\_cloglog[3] 8.000000e-02 0.000000e+00 0.000000e+00 7.000000e-02 8.000000e-02 8.000000e-02 9.000000e-02 9.000000e-02 105647 1.00

p\_cloglog[4] 3.000000e-02 0.000000e+00 0.000000e+00 2.000000e-02 2.000000e-02 3.000000e-02 3.000000e-02 3.000000e-02 111399 1.00

p\_cloglog[5] 4.000000e-02 0.000000e+00 0.000000e+00 4.000000e-02 4.000000e-02 4.000000e-02 4.000000e-02 4.000000e-02 104556 1.00

p\_cloglog[6] 6.000000e-02 0.000000e+00 0.000000e+00 6.000000e-02 6.000000e-02 6.000000e-02 7.000000e-02 7.000000e-02 103682 1.00

p\_cloglog[7] 6.000000e-02 0.000000e+00 0.000000e+00 6.000000e-02 6.000000e-02 6.000000e-02 6.000000e-02 7.000000e-02 103105 1.00

p\_cloglog[8] 4.000000e-02 0.000000e+00 1.000000e-02 2.000000e-02 3.000000e-02 4.000000e-02 5.000000e-02 6.000000e-02 100385 1.00

p\_cloglog[9] 3.000000e-02 0.000000e+00 0.000000e+00 3.000000e-02 3.000000e-02 3.000000e-02 4.000000e-02 4.000000e-02 99613 1.00

p\_cloglog[10] 1.800000e-01 0.000000e+00 1.000000e-02 1.700000e-01 1.800000e-01 1.800000e-01 1.800000e-01 1.900000e-01 105421 1.00

p\_cloglog[11] 5.000000e-02 0.000000e+00 0.000000e+00 4.000000e-02 4.000000e-02 5.000000e-02 5.000000e-02 5.000000e-02 101228 1.00

p\_cloglog[12] 1.100000e-01 0.000000e+00 0.000000e+00 1.000000e-01 1.000000e-01 1.100000e-01 1.100000e-01 1.100000e-01 106274 1.00

[ reached getOption("max.print") -- omitted 146 rows ]

Samples were drawn using NUTS(diag\_e) at Thu Aug 29 16:22:25 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]", "d[3]", "d[4]", "d[5]", "d[6]", "T[1]", "T[2]", "T[3]", "T[4]", "T[5]", "T[6]", "totalresdev"), probs = c(0.025, 0.975))$summary

> Stan\_summary

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

d[2] -1.960184e-03 0.003064765 0.9913371 -1.940992 1.9337013 104628.13 0.9999436

d\_II[2] -1.960184e-03 0.003064765 0.9913371 -1.940992 1.9337013 104628.13 0.9999436

d[3] 4.166617e-03 0.002968276 0.9988155 -1.944933 1.9565967 113230.12 0.9999753

d[4] 1.051165e-03 0.003029689 1.0008046 -1.946147 1.9436078 109119.57 0.9999606

d[5] 2.851480e-03 0.003101799 1.0050326 -1.948545 1.9702678 104986.41 0.9999490

d[6] 8.016117e-04 0.003119100 1.0094333 -1.974833 1.9887254 104736.17 0.9999615

T[1] -3.102265e+00 0.002943152 0.9517609 -4.961025 -1.2524339 104575.62 0.9999781

T[2] -3.103711e+00 0.004316873 1.3744001 -5.786274 -0.4113876 101364.95 0.9999647

T[3] -3.093880e+00 0.004243930 1.3907147 -5.807516 -0.3685007 107384.04 0.9999435

T[4] -3.102359e+00 0.004083515 1.3712846 -5.764093 -0.4310797 112768.29 0.9999482

T[5] -3.102219e+00 0.004222867 1.3867594 -5.796506 -0.3497571 107841.85 0.9999574

T[6] -3.099864e+00 0.004275512 1.3788718 -5.818608 -0.3663716 104009.12 0.9999545

totalresdev 2.207601e+02 0.054352797 8.8705016 205.555486 240.1577992 26634.99 0.9999836

>

>

>

> ##### Additional Lines for Median

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

> Median\_d2

[1] 0.0003253434

> Median\_d3 <- median(as.matrix(stan\_samples, pars = c("d\_II[3]")))

> Median\_d3

[1] 0.006486178

> Median\_d4 <- median(as.matrix(stan\_samples, pars = c("d\_II[4]")))

> Median\_d4

[1] 0.004924392

> Median\_d5 <- median(as.matrix(stan\_samples, pars = c("d\_II[5]")))

> Median\_d5

[1] -0.00260456

> Median\_d6 <- median(as.matrix(stan\_samples, pars = c("d\_II[6]")))

> Median\_d6

[1] -0.0002951853

>

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

> Median\_T1

[1] -3.10117

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

> Median\_T2

[1] -3.106752

> Median\_T3 <- median(as.matrix(stan\_samples, pars = c("T[3]")))

> Median\_T3

[1] -3.104045

> Median\_T4 <- median(as.matrix(stan\_samples, pars = c("T[4]")))

> Median\_T4

[1] -3.100774

> Median\_T5 <- median(as.matrix(stan\_samples, pars = c("T[5]")))

> Median\_T5

[1] -3.102189

> Median\_T6 <- median(as.matrix(stan\_samples, pars = c("T[6]")))

> Median\_T6

[1] -3.096798

>

>

>

>

>

> # 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. : 0.005 Min. : 0.0 Min. : 1 Min. :0.00000 Min. : 38453

1st Qu.:0.73 1st Qu.: 0.590 1st Qu.: 3.0 1st Qu.: 7 1st Qu.:0.00000 1st Qu.: 38469

Median :0.87 Median : 0.624 Median : 3.0 Median : 7 Median :0.00000 Median : 38473

Mean :0.83 Mean : 0.634 Mean : 3.6 Mean : 69 Mean :0.00059 Mean : 38511

3rd Qu.:0.97 3rd Qu.: 0.678 3rd Qu.: 3.0 3rd Qu.: 7 3rd Qu.:0.00000 3rd Qu.: 38477

Max. :1.00 Max. :14.386 Max. :10.0 Max. :1023 Max. :1.00000 Max. :512391

> # each chain separately

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

[[1]]

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

Min. :0.00 Min. : 0.005 Min. : 0.0 Min. : 1 Min. :0.00000 Min. : 38454

1st Qu.:0.73 1st Qu.: 0.590 1st Qu.: 3.0 1st Qu.: 7 1st Qu.:0.00000 1st Qu.: 38469

Median :0.87 Median : 0.590 Median : 3.0 Median : 7 Median :0.00000 Median : 38473

Mean :0.83 Mean : 0.637 Mean : 3.6 Mean : 70 Mean :0.00063 Mean : 38511

3rd Qu.:0.97 3rd Qu.: 0.679 3rd Qu.: 3.0 3rd Qu.: 7 3rd Qu.:0.00000 3rd Qu.: 38477

Max. :1.00 Max. :14.386 Max. :10.0 Max. :1023 Max. :1.00000 Max. :511761

[[2]]

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

Min. :0.00 Min. : 0.0062 Min. : 0.0 Min. : 1 Min. :0e+00 Min. : 38453

1st Qu.:0.75 1st Qu.: 0.5486 1st Qu.: 3.0 1st Qu.: 7 1st Qu.:0e+00 1st Qu.: 38469

Median :0.88 Median : 0.5486 Median : 3.0 Median : 7 Median :0e+00 Median : 38473

Mean :0.84 Mean : 0.6148 Mean : 3.6 Mean : 68 Mean :6e-04 Mean : 38511

3rd Qu.:0.97 3rd Qu.: 0.6783 3rd Qu.: 3.0 3rd Qu.: 7 3rd Qu.:0e+00 3rd Qu.: 38477

Max. :1.00 Max. :14.3855 Max. :10.0 Max. :1023 Max. :1e+00 Max. :506139

[[3]]

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

Min. :0.00 Min. : 0.0052 Min. : 0.0 Min. : 1 Min. :0.00000 Min. : 38454

1st Qu.:0.71 1st Qu.: 0.6243 1st Qu.: 3.0 1st Qu.: 7 1st Qu.:0.00000 1st Qu.: 38469

Median :0.86 Median : 0.6243 Median : 3.0 Median : 7 Median :0.00000 Median : 38473

Mean :0.82 Mean : 0.6510 Mean : 3.6 Mean : 69 Mean :0.00053 Mean : 38511

3rd Qu.:0.96 3rd Qu.: 0.6761 3rd Qu.: 3.0 3rd Qu.: 7 3rd Qu.:0.00000 3rd Qu.: 38477

Max. :1.00 Max. :14.3855 Max. :10.0 Max. :1023 Max. :1.00000 Max. :512391

> 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\_\_ -38455.96 0.03277337 4.061625 -38464.89 -38449.02 15358.8 0.999989

> # 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]" , "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]", "dev[45]", "dev[46]", "dev[47]", "dev[48]"))$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 48 Werte

> #dev ist Std-Abweichung jedes einzelnen Werts

> # insg 48 Werte

> Var\_manuell <- sum(Hilf\_dev\_pD[,2])^2/48

> Var\_manuell

[1] 23330094101

> pD\_manuell <- Var\_manuell/2

> pD\_manuell

[1] 11665047051

>

> #Var\_manuell <- sum(Hilf\_dev\_pD[,4])/48

> #Var\_manuell

>

>

>

>

>

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