

# Real data analysis report

Yunfeng Yang

5/25/2021

```
require(rstan)
require(MASS)
require(loo)
library(readxl)
library(dplyr)
```

## Data preprocessing (Please skip this section)

```
####-----data preprocessing-----
#setwd("~/Downloads/github/Bayesian-lumber-strength/R/Real data analysis")

summary_all08122013 <- read_excel("summary_all08122013.xlsx")

summary_all08122013 <- select(summary_all08122013, "Group", "Broken", "MOR", "UTS")

## T group
T100_data <- summary_all08122013[which(summary_all08122013$Group == "T100"),]
T60_data <- summary_all08122013[which(summary_all08122013$Group == "T60"),]
T40_data <- summary_all08122013[which(summary_all08122013$Group == "T40"),]
T20_data <- summary_all08122013[which(summary_all08122013$Group == "T20"),]
T100_data$UTS <- as.numeric(T100_data$UTS)
T100_data <- T100_data$UTS
## G group
R100_data <- summary_all08122013[which(summary_all08122013$Group == "R100"),]
R60_data <- summary_all08122013[which(summary_all08122013$Group == "R60"),]
R40_data <- summary_all08122013[which(summary_all08122013$Group == "R40"),]
R20_data <- summary_all08122013[which(summary_all08122013$Group == "R20"),]
R100_data <- as.numeric(R100_data$MOR)

##-----T substitute NA to 0 -----

id <- T60_data$Broken == 1
T60_data$MOR[id] <- 0
T60_data$UTS[id] <- as.numeric(T60_data$UTS[id])
T60_data$UTS[!id] <- 0
T60_data$UTS <- as.numeric(T60_data$UTS)
T60_data$MOR <- as.numeric(T60_data$MOR)
```

```

id <- T40_data$Broken == 1
T40_data$MOR[id] <- 0
T40_data$UTS[id] <- as.numeric(T40_data$UTS[id])
T40_data$UTS[!id] <- 0
T40_data$UTS <- as.numeric(T40_data$UTS)
T40_data$MOR <- as.numeric(T40_data$MOR)

id <- T20_data$Broken == 1
T20_data$MOR[id] <- 0
T20_data$UTS[id] <- as.numeric(T20_data$UTS[id])
T20_data$UTS[!id] <- 0
T20_data$UTS <- as.numeric(T20_data$UTS)
T20_data$MOR <- as.numeric(T20_data$MOR)

##-----R substitute NA to 0 -----

id <- R60_data$Broken == 0
R60_data$MOR[id] <- 0
R60_data$UTS[id] <- as.numeric(R60_data$UTS[id])
R60_data$UTS[!id] <- 0
R60_data$UTS <- as.numeric(R60_data$UTS)
R60_data$MOR <- as.numeric(R60_data$MOR)

id <- R40_data$Broken == 0
R40_data$MOR[id] <- 0
R40_data$UTS[id] <- as.numeric(R40_data$UTS[id])
R40_data$UTS[!id] <- 0
R40_data$UTS <- as.numeric(R40_data$UTS)
R40_data$MOR <- as.numeric(R40_data$MOR)

id <- R20_data$Broken == 0
R20_data$MOR[id] <- 0
R20_data$UTS[id] <- as.numeric(R20_data$UTS[id])
R20_data$UTS[!id] <- 0
R20_data$UTS <- as.numeric(R20_data$UTS)
R20_data$MOR <- as.numeric(R20_data$MOR)

##-----Convert psi to Mpa-----

# 1 thousand psi = 6.895 MPa

c = 6.895

R20_data$MOR <- c*R20_data$MOR
R20_data$UTS <- c*R20_data$UTS

```

```

R40_data$MOR <- c*R40_data$MOR
R40_data$UTS <- c*R40_data$UTS

R60_data$MOR <- c*R60_data$MOR
R60_data$UTS <- c*R60_data$UTS

R100_data <- c*R100_data

T20_data$MOR <- c*T20_data$MOR
T20_data$UTS <- c*T20_data$UTS

T40_data$MOR <- c*T40_data$MOR
T40_data$UTS <- c*T40_data$UTS

T60_data$MOR <- c*T60_data$MOR
T60_data$UTS <- c*T60_data$UTS

T100_data <- c*T100_data

##-----check normal fitting-----
#
# library(fitdistrplus)
#
# # check R100_data
# FIT <- fitdist(R100_data, "norm")    ## note: it is "norm" not "normal"
# plot(FIT)    ## use method `plot.fitdist`
# FIT$estimate # mean = 45.679 sd = 12.900
# FIT$bic # bic = 1394.022
# # good normal fitting
#
#
# # check T100_data
#
# shapiro.test(T100_data) # p = 0.0001841
# shapiro.test(log(T100_data)) # p-value = 0.002327
# shapiro.test(sqrt(T100_data)) # p-value = 0.3988
#


##-----proof loading-----
R_pf <- c* c(4.956690733, 6.110714122, 7.092435407)
T_pf <- sqrt(c* c(2.962390379, 3.986497991, 4.916102264))

##-----convert UTS to sqrt(UTS)

R20_data$UTS <- sqrt(R20_data$UTS)
R40_data$UTS <- sqrt(R40_data$UTS)
R60_data$UTS <- sqrt(R60_data$UTS)
R100_data <- R100_data

```

```

T20_data$UTS <- sqrt(T20_data$UTS)
T40_data$UTS <- sqrt(T40_data$UTS)
T60_data$UTS <- sqrt(T60_data$UTS)
T100_data <- sqrt(T100_data)

##-----check outlier and delete it-----

# R20
id <- which((R20_data$MOR < R_pf[1]) == F)
R20_data <- R20_data[-id,]

# R40
id <- which((R40_data$MOR < R_pf[2]) == F)

# R60
id <- which((R60_data$MOR < R_pf[3]) == F)
R60_data <- R60_data[-id,]

# T20
id <- which((T20_data$UTS < T_pf[1]) == F)
T20_data <- T20_data[-id,]

# T40
id <- which((T40_data$UTS < T_pf[2]) == F)
T40_data <- T40_data[-id,]

# T60
id <- which((T60_data$UTS < T_pf[3]) == F)
T60_data <- T60_data[-id,]

##---- Convert the data so stan can use it ----
R20_data <- cbind(R20_data$MOR,R20_data$UTS,R20_data$Broken)
R40_data <- cbind(R40_data$MOR,R40_data$UTS,R40_data$Broken)
R60_data <- cbind(R60_data$MOR,R60_data$UTS,R60_data$Broken)
T20_data <- cbind(T20_data$UTS,T20_data$MOR,T20_data$Broken)
T40_data <- cbind(T40_data$UTS,T40_data$MOR,T40_data$Broken)
T60_data <- cbind(T60_data$UTS,T60_data$MOR,T60_data$Broken)

```

## Model 1: include all alpha's

```

dmg_mod <- stan_model("damage.stan")

## Trying to compile a simple C file
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.frame
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/inc

```

```

## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util
## namespace Eigen {
## ^
## ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: f
## #include <complex>
## ^
## 3 errors generated.
## make: *** [foo.o] Error 1

init_dmg <- function() {
  list(mu = c(35,8), sigma = c(10,1), rho = .5, alpha_R20 = 1,
       alpha_R40 = 1, alpha_R60 = 1, alpha_T20 = 1, alpha_T40 = 1, alpha_T60 = 1 )
}

set.seed(2020)
dmg_fit <- sampling(object = dmg_mod,
                      data = list(N_R20 = nrow(R20_data), N_R40 = nrow(R40_data), N_R60 = nrow(R60_data),
                                  N_T20 = nrow(T20_data), N_T40 = nrow(T40_data), N_T60 = nrow(T60_data),
                                  N_x = length(T100_data), N_y = length(R100_data),
                                  X_R20 = R20_data, X_R40 = R40_data, X_R60 = R60_data,
                                  X_T20 = T20_data, X_T40 = T40_data, X_T60 = T60_data,
                                  t_x = R100_data, t_y = T100_data,
                                  l_R20=R_pf[1], l_R40=R_pf[2], l_R60=R_pf[3],
                                  l_T20=T_pf[1], l_T40=T_pf[2], l_T60=T_pf[3]),
                      control = list(adapt_delta = 0.8), init = init_dmg, refresh = 0)

print(dmg_fit, pars = c('mu', 'sigma', 'rho', 'alpha_R20', 'alpha_R40',
                       'alpha_R60', 'alpha_T20', 'alpha_T40', 'alpha_T60'))

## Inference for Stan model: damage.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean    sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## mu[1]    45.13    0.01  0.59 43.94 44.74 45.13 45.53 46.31  3450     1
## mu[2]     5.51    0.00  0.05  5.41  5.48  5.51  5.55  5.62  3103     1
## sigma[1]  12.92    0.01  0.44 12.09 12.62 12.91 13.22 13.85  3605     1
## sigma[2]   1.08    0.00  0.04  1.01  1.05  1.08  1.10  1.16  3487     1
## rho       0.80    0.00  0.05  0.70  0.77  0.80  0.83  0.88  2437     1
## alpha_R20  0.65    0.01  0.47  0.03  0.28  0.57  0.93  1.75  3132     1
## alpha_R40  1.80    0.02  0.85  0.24  1.20  1.76  2.35  3.60  1884     1
## alpha_R60  1.28    0.02  0.83  0.10  0.63  1.16  1.78  3.16  2444     1
## alpha_T20  84.10   0.92 53.41  5.38 41.97 76.90 118.22 207.64  3405     1
## alpha_T40 102.25   1.21 65.99  6.25 50.31 92.92 143.94 252.53  2983     1
## alpha_T60  65.37   0.89 55.13  1.87 22.90 51.19  93.78 208.22  3798     1
##
## Samples were drawn using NUTS(diag_e) at Tue May 25 14:49:48 2021.

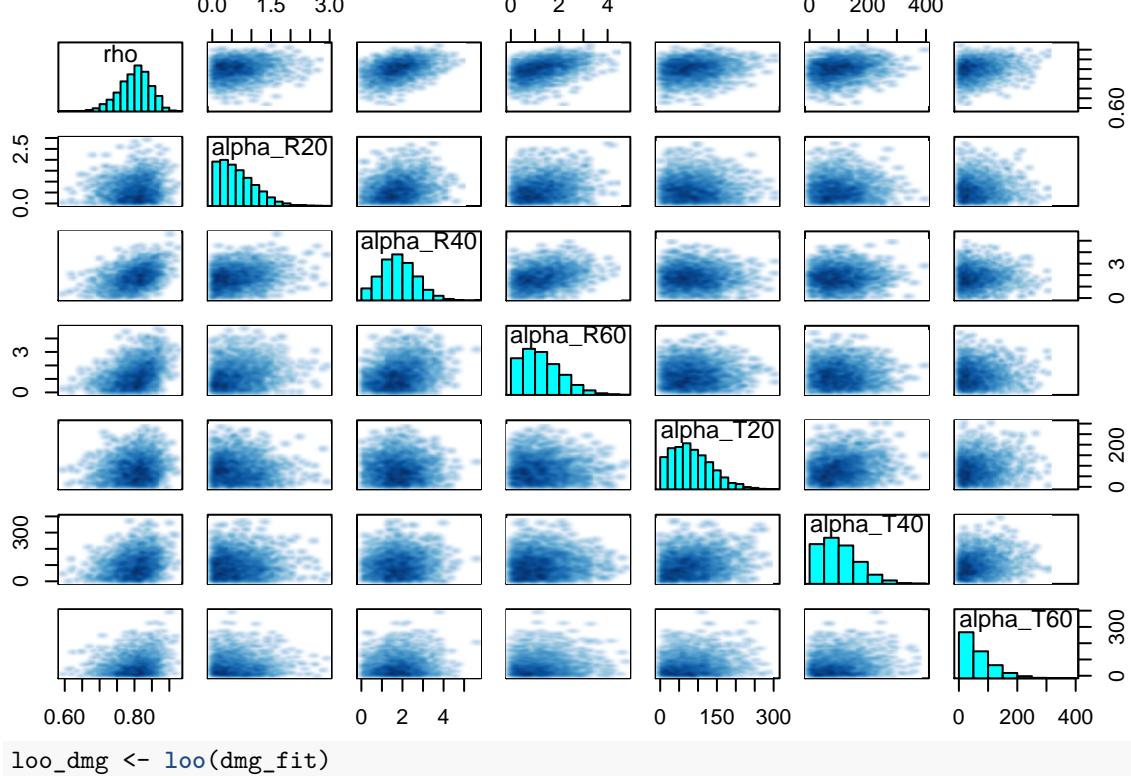
```

```

## 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).

pairs(dmg_fit,pars = c('rho','alpha_R20','alpha_R40',
                      'alpha_R60','alpha_T20','alpha_T40','alpha_T60'))

```



```
loo_dmg <- loo(dmg_fit)
```

## Model 2: include no alpha's, no damage

```
nondmg_mod <- stan_model("nondamage.stan")
```

```

## Trying to compile a simple C file
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include"
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util
## namespace Eigen {
## ^
## ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include

```

```

## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: f
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: f
## #include <complex>
##
## ~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1

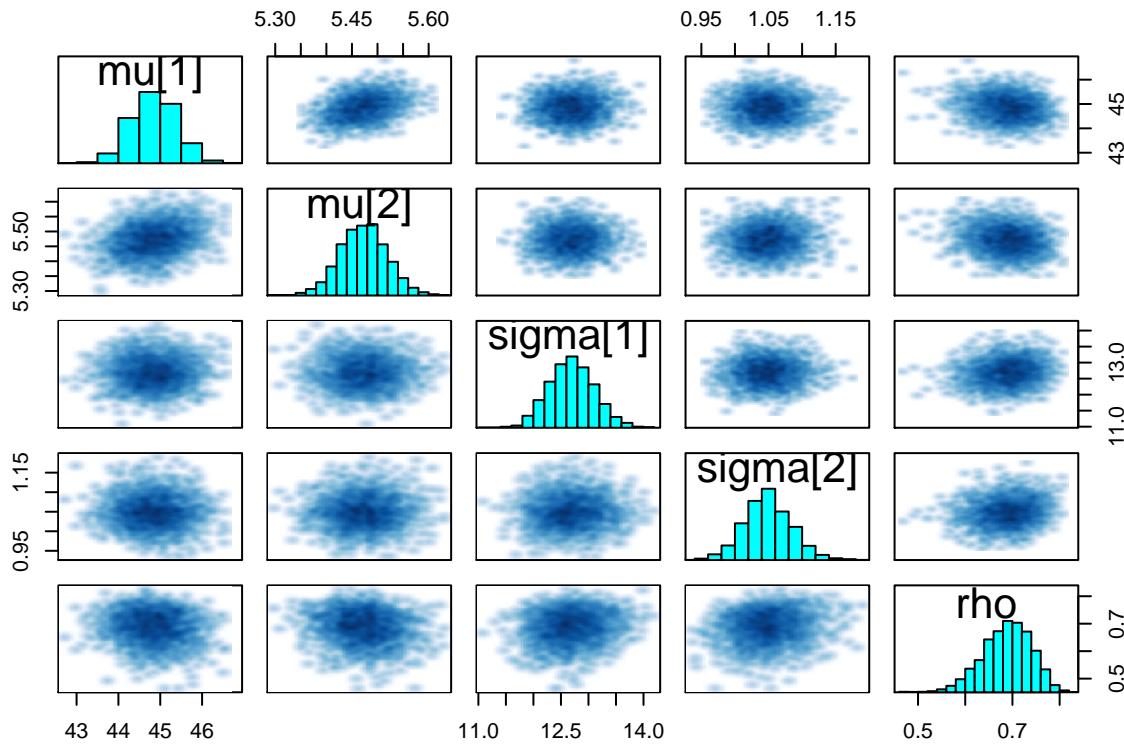
init_nondmg <- function() {
  list(mu = c(35,8), sigma = c(10,1), rho = .5)
}

nondmg_fit <- sampling(object = nondmg_mod,
                        data = list(N_R20 = nrow(R20_data), N_R40 = nrow(R40_data), N_R60 = nrow(R60_data),
                                    N_T20 = nrow(T20_data), N_T40 = nrow(T40_data), N_T60 = nrow(T60_data),
                                    N_x = length(T100_data), N_y = length(R100_data),
                                    X_R20 = R20_data, X_R40 = R40_data, X_R60 = R60_data,
                                    X_T20 = T20_data, X_T40 = T40_data, X_T60 = T60_data,
                                    t_x = R100_data, t_y = T100_data,
                                    l_R20=R_pf[1], l_R40=R_pf[2], l_R60=R_pf[3],
                                    l_T20=T_pf[1], l_T40=T_pf[2], l_T60=T_pf[3]),
                        control = list(adapt_delta = 0.8), init = init_nondmg, refresh = 0)
print(nondmg_fit, pars = c('mu', 'sigma', 'rho'))

## Inference for Stan model: nondamage.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean    sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## mu[1]    44.85     0.01 0.54 43.83 44.47 44.85 45.21 45.89  4266     1
## mu[2]     5.47     0.00 0.05  5.38  5.44  5.47  5.50  5.56  3816     1
## sigma[1] 12.69     0.01 0.41 11.92 12.41 12.68 12.97 13.52  4590     1
## sigma[2]  1.05     0.00 0.04  0.98  1.03  1.05  1.07  1.12  3444     1
## rho       0.69     0.00 0.05  0.58  0.65  0.69  0.73  0.78  3948     1
##
## Samples were drawn using NUTS(diag_e) at Tue May 25 14:51:10 2021.
## 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).

pairs(nondmg_fit, pars = c('mu', 'sigma', 'rho'))

```



```
# LOOIC
loo_nondamage <- loo(nondmg_fit)
```

### model 3: include only positive alpha\_R40

```
R40dmg_mod <- stan_model("only_alphaR40.stan")

## Trying to compile a simple C file
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include"
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util
## namespace Eigen {
##   ~
##   /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util
##   namespace Eigen {
##   ^
##   ;
##   In file included from <built-in>:1:
##   In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include
##   In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include
##   /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: f
##   #include <complex>
##   ^~~~~~
```

```

## 3 errors generated.
## make: *** [foo.o] Error 1

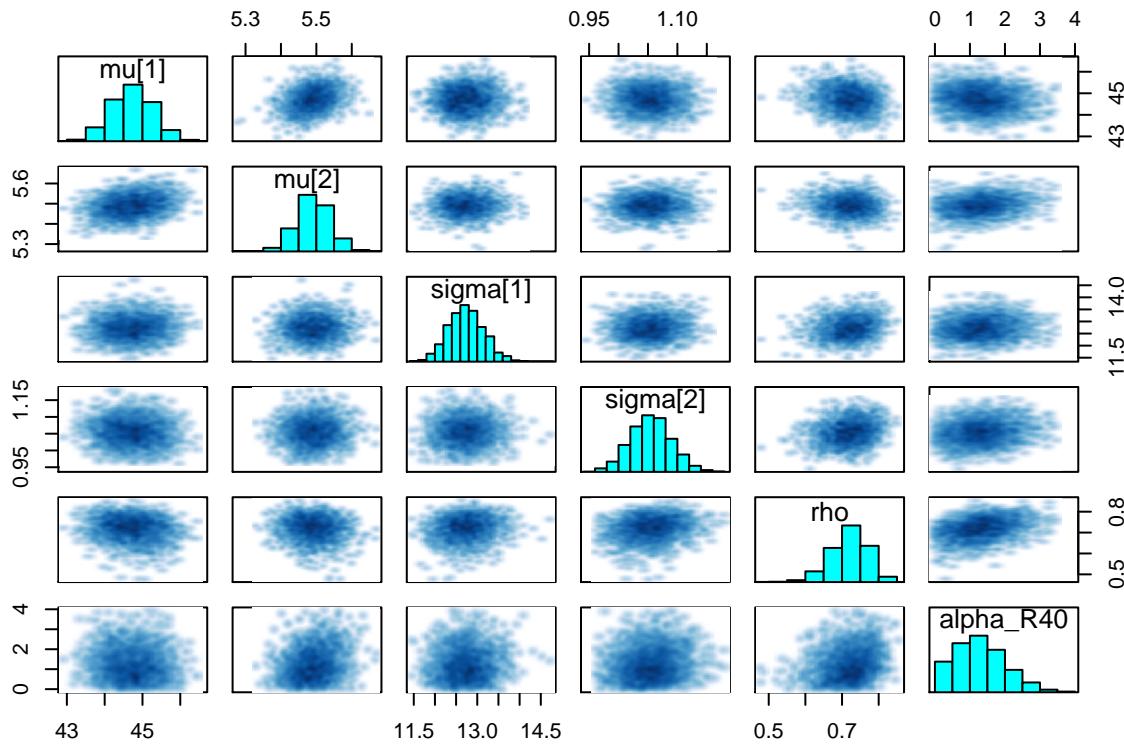
init_R40dmg <- function() {
  list(mu = c(35,8), sigma = c(10,1), rho = .5, alpha_R40 = 1)
}

R40dmg_fit <- sampling(object = R40dmg_mod,
                        data = list(N_R20 = nrow(R20_data), N_R40 = nrow(R40_data), N_R60 = nrow(R60_data),
                                    N_T20 = nrow(T20_data), N_T40 = nrow(T40_data), N_T60 = nrow(T60_data),
                                    N_x = length(T100_data), N_y = length(R100_data),
                                    X_R20 = R20_data, X_R40 = R40_data, X_R60 = R60_data,
                                    X_T20 = T20_data, X_T40 = T40_data, X_T60 = T60_data,
                                    t_x = R100_data, t_y = T100_data,
                                    l_R20=R_pf[1],l_R40=R_pf[2],l_R60=R_pf[3],
                                    l_T20=T_pf[1],l_T40=T_pf[2],l_T60=T_pf[3]),
                        control = list(adapt_delta = 0.8), init = init_R40dmg, refresh = 0)
print(R40dmg_fit,pars = c('mu','sigma','rho','alpha_R40'))

## Inference for Stan model: only_alphaR40.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean    sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## mu[1]    44.73    0.01 0.54 43.70 44.35 44.72 45.11 45.76  4171     1
## mu[2]     5.49    0.00 0.05  5.40  5.46  5.49  5.52  5.58  3988     1
## sigma[1]  12.74    0.01 0.42 11.96 12.44 12.73 13.01 13.62  4284     1
## sigma[2]   1.06    0.00 0.03  0.99  1.03  1.05  1.08  1.12  4203     1
## rho       0.72    0.00 0.05  0.61  0.69  0.72  0.76  0.81  3646     1
## alpha_R40 1.30    0.01 0.72  0.13  0.76  1.24  1.77  2.84  2659     1
##
## Samples were drawn using NUTS(diag_e) at Tue May 25 14:52:33 2021.
## 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).

pairs(R40dmg_fit,pars = c('mu','sigma','rho','alpha_R40'))

```



```
# LOOIC
```

```
loo_R40dmg <- loo(R40dmg_fit)
```

## Model 4: include only real alpha\_R40

```
R40dmgreal_mod <- stan_model("only_alphaR40_real.stan")
```

```
## Trying to compile a simple C file
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/...
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/...
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/...
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/...
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util...
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util...
## namespace Eigen {
## ^
## ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/...
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/...
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: f...
```

```

## 3 errors generated.
## make: *** [foo.o] Error 1

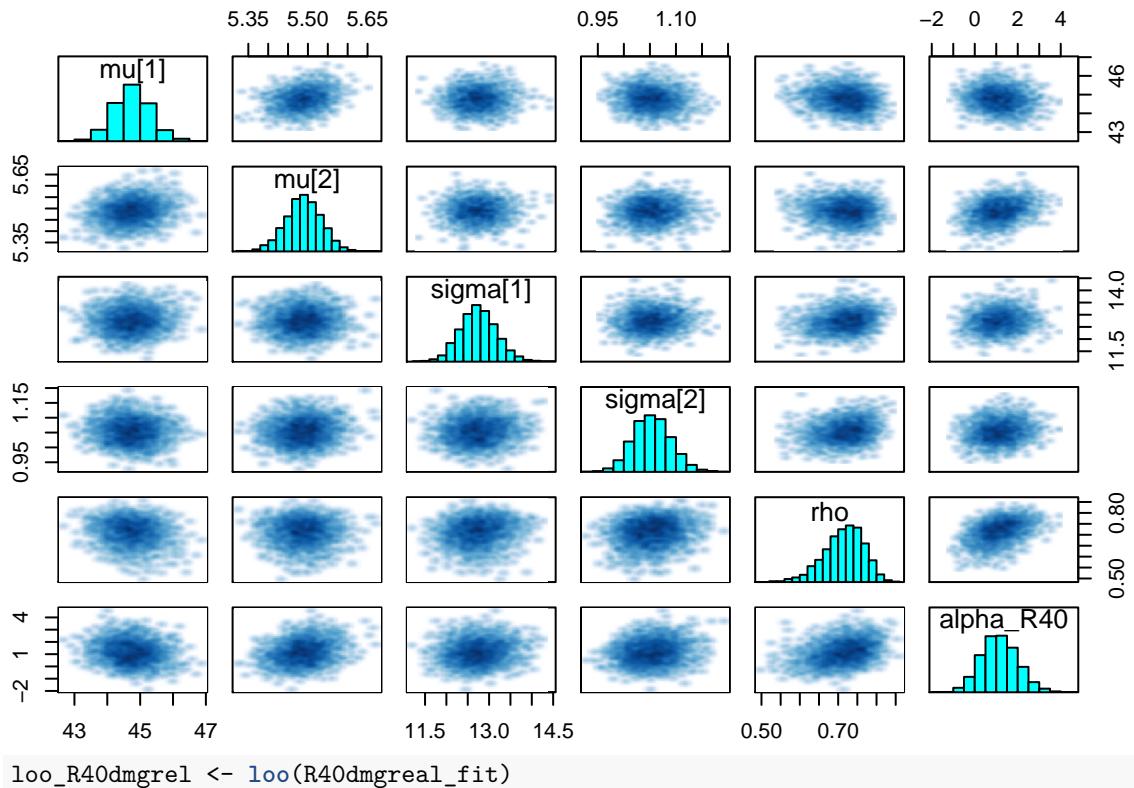
init_R40dmg <- function() {
  list(mu = c(35,8), sigma = c(10,1), rho = .5, alpha_R40 = 1)
}

R40dmgreal_fit <- sampling(object = R40dmgreal_mod,
                             data = list(N_R20 = nrow(R20_data), N_R40 = nrow(R40_data), N_R60 = nrow(R60_data),
                                         N_T20 = nrow(T20_data), N_T40 = nrow(T40_data), N_T60 = nrow(T60_data),
                                         N_x = length(T100_data), N_y = length(R100_data),
                                         X_R20 = R20_data, X_R40 = R40_data, X_R60 = R60_data,
                                         X_T20 = T20_data, X_T40 = T40_data, X_T60 = T60_data,
                                         t_x = R100_data, t_y = T100_data,
                                         l_R20=R_pf[1],l_R40=R_pf[2],l_R60=R_pf[3],
                                         l_T20=T_pf[1],l_T40=T_pf[2],l_T60=T_pf[3]),
                             control = list(adapt_delta = 0.8), init = init_R40dmg, refresh = 0)
print(R40dmgreal_fit,pars = c('mu','sigma','rho','alpha_R40'))

## Inference for Stan model: only_alphaR40_real.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean    sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## mu[1]    44.76    0.01 0.56 43.67 44.38 44.75 45.12 45.90  5054     1
## mu[2]     5.49    0.00 0.05  5.40  5.46  5.49  5.52  5.58  5105     1
## sigma[1]  12.75    0.01 0.43 11.94 12.46 12.74 13.03 13.63  5298     1
## sigma[2]   1.05    0.00 0.04  0.99  1.03  1.05  1.08  1.13  5661     1
## rho       0.71    0.00 0.05  0.60  0.68  0.72  0.75  0.80  4570     1
## alpha_R40 1.15    0.01 0.84 -0.41  0.56  1.11  1.72  2.87  4567     1
##
## Samples were drawn using NUTS(diag_e) at Tue May 25 14:53:44 2021.
## 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).

pairs(R40dmgreal_fit,pars = c('mu','sigma','rho','alpha_R40'))

```



## Compare LOOIC across the 4 models

```

rbind(allalpha = loo_dmg$estimates[3,1],
      noalpha = loo_nondamage$estimates[3,1],
      positiveR40 = loo_R40dmg$estimates[3,1],
      realR40 = loo_R40dmgrel$estimates[3,1])

##          [,1]
## allalpha    4975.535
## noalpha    4969.043
## positiveR40 4968.508
## realR40    4969.163

```