```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.78 (8075) aarch64-apple-darwin20]

> ## Started 10 October 2024 ##
> ## By Lizzie ##
> 
> ## Following analyses_compmod_exp2.R some ##
> ## Using Stan and loo to do similar model comparison as Isabelle did in Scripts__expe2_model_comparison.R ##
> 
> ## See https://mc-stan.org/loo/articles/loo2-example.html for info on loo ##
> ## I have not fully reviewed the above page, I am mostly just following what I did in analyses_compmod_exp2.R ##
> 
> rm(list=ls()) # remove everything currently held in the R memory
> options(stringsAsFactors=FALSE)
> 
> # Load libraries
> library(rstanarm)
Loading required package: Rcpp
This is rstanarm version 2.21.3
- See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
- Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
- For execution on a local, multicore CPU with excess RAM we recommend calling
  options(mc.cores = parallel::detectCores())
> 
> setwd("~/Documents/git/projects/treegarden/misc/isabelle_expe/modelcompare")
> 
> d <- read.delim("input/data_expe2_v5.csv", header=TRUE, sep=";")
> acer <- subset(d, Species=="Acer")
> 
> # Run all the single predictor models
> acer.null <- stan_lmer(BBdelay ~ (1|Tree_ID/Cutting_code), data=acer, cores=4)
> acer.day <- stan_lmer(BBdelay ~ DayT + (1|Tree_ID/Cutting_code), data=acer, cores=4)
> acer.night <- stan_lmer(BBdelay ~ NightT + (1|Tree_ID/Cutting_code), data=acer, cores=4)
> acer.max <- stan_lmer(BBdelay ~ MaxT + (1|Tree_ID/Cutting_code), data=acer, cores=4)
> acer.min <- stan_lmer(BBdelay ~ MinT + (1|Tree_ID/Cutting_code), data=acer, cores=4)
> acer.dawn <- stan_lmer(BBdelay ~ DawnT + (1|Tree_ID/Cutting_code), data=acer, cores=4)
> acer.dusk <- stan_lmer(BBdelay ~ DuskT + (1|Tree_ID/Cutting_code), data=acer, cores=4)
> acer.diff <- stan_lmer(BBdelay ~ DIFF + (1|Tree_ID/Cutting_code), data=acer, cores=4)
> acer.mean <- stan_lmer(BBdelay ~  MeanT + (1|Tree_ID/Cutting_code), data=acer, cores=4)
> 
> loo.acer.null <- loo(acer.null, save_psis = TRUE, k_threshold = 0.7)
10 problematic observation(s) found.
Model will be refit 10 times.

Fitting model 1 out of 10 (leaving out observation 32)

Fitting model 2 out of 10 (leaving out observation 100)

Fitting model 3 out of 10 (leaving out observation 105)

Fitting model 4 out of 10 (leaving out observation 187)

Fitting model 5 out of 10 (leaving out observation 198)

Fitting model 6 out of 10 (leaving out observation 199)

Fitting model 7 out of 10 (leaving out observation 247)

Fitting model 8 out of 10 (leaving out observation 252)

Fitting model 9 out of 10 (leaving out observation 318)

Fitting model 10 out of 10 (leaving out observation 319)
> loo.acer.day <- loo(acer.day, save_psis = TRUE, k_threshold = 0.7)
12 problematic observation(s) found.
Model will be refit 12 times.

Fitting model 1 out of 12 (leaving out observation 31)
Fitting model 2 out of 12 (leaving out observation 32)

Fitting model 3 out of 12 (leaving out observation 98)

Fitting model 4 out of 12 (leaving out observation 100)

Fitting model 5 out of 12 (leaving out observation 105)

Fitting model 6 out of 12 (leaving out observation 144)

Fitting model 7 out of 12 (leaving out observation 187)

Fitting model 8 out of 12 (leaving out observation 198)

Fitting model 9 out of 12 (leaving out observation 199)

Fitting model 10 out of 12 (leaving out observation 252)

Fitting model 11 out of 12 (leaving out observation 255)

Fitting model 12 out of 12 (leaving out observation 264)
> loo.acer.night <- loo(acer.night, save_psis = TRUE, k_threshold = 0.7)
11 problematic observation(s) found.
Model will be refit 11 times.

Fitting model 1 out of 11 (leaving out observation 32)

Fitting model 2 out of 11 (leaving out observation 100)

Fitting model 3 out of 11 (leaving out observation 105)

Fitting model 4 out of 11 (leaving out observation 144)

Fitting model 5 out of 11 (leaving out observation 195)

Fitting model 6 out of 11 (leaving out observation 198)

Fitting model 7 out of 11 (leaving out observation 199)

Fitting model 8 out of 11 (leaving out observation 246)

Fitting model 9 out of 11 (leaving out observation 247)

Fitting model 10 out of 11 (leaving out observation 255)

Fitting model 11 out of 11 (leaving out observation 264)
> loo.acer.max <- loo(acer.max, save_psis = TRUE, k_threshold = 0.7)
12 problematic observation(s) found.
Model will be refit 12 times.

Fitting model 1 out of 12 (leaving out observation 31)

Fitting model 2 out of 12 (leaving out observation 33)

Fitting model 3 out of 12 (leaving out observation 98)

Fitting model 4 out of 12 (leaving out observation 100)

Fitting model 5 out of 12 (leaving out observation 105)

Fitting model 6 out of 12 (leaving out observation 144)

Fitting model 7 out of 12 (leaving out observation 195)

Fitting model 8 out of 12 (leaving out observation 198)

Fitting model 9 out of 12 (leaving out observation 199)

Fitting model 10 out of 12 (leaving out observation 247)

Fitting model 11 out of 12 (leaving out observation 252)

Fitting model 12 out of 12 (leaving out observation 255)
> loo.acer.min <- loo(acer.min, save_psis = TRUE, k_threshold = 0.7)
11 problematic observation(s) found.
Model will be refit 11 times.

Fitting model 1 out of 11 (leaving out observation 32)

Fitting model 2 out of 11 (leaving out observation 96)

Fitting model 3 out of 11 (leaving out observation 99)

Fitting model 4 out of 11 (leaving out observation 100)

Fitting model 5 out of 11 (leaving out observation 105)
Fitting model 6 out of 11 (leaving out observation 144)

Fitting model 7 out of 11 (leaving out observation 195)

Fitting model 8 out of 11 (leaving out observation 198)

Fitting model 9 out of 11 (leaving out observation 199)

Fitting model 10 out of 11 (leaving out observation 247)

Fitting model 11 out of 11 (leaving out observation 252)
> loo.acer.dawn <- loo(acer.dawn, save_psis = TRUE, k_threshold = 0.7)
12 problematic observation(s) found.
Model will be refit 12 times.

Fitting model 1 out of 12 (leaving out observation 32)

Fitting model 2 out of 12 (leaving out observation 58)

Fitting model 3 out of 12 (leaving out observation 96)

Fitting model 4 out of 12 (leaving out observation 100)

Fitting model 5 out of 12 (leaving out observation 105)

Fitting model 6 out of 12 (leaving out observation 186)

Fitting model 7 out of 12 (leaving out observation 198)

Fitting model 8 out of 12 (leaving out observation 199)

Fitting model 9 out of 12 (leaving out observation 247)

Fitting model 10 out of 12 (leaving out observation 255)

Fitting model 11 out of 12 (leaving out observation 305)

Fitting model 12 out of 12 (leaving out observation 317)
> loo.acer.dusk <- loo(acer.dusk, save_psis = TRUE, k_threshold = 0.7)
13 problematic observation(s) found.
Model will be refit 13 times.

Fitting model 1 out of 13 (leaving out observation 33)

Fitting model 2 out of 13 (leaving out observation 91)

Fitting model 3 out of 13 (leaving out observation 99)

Fitting model 4 out of 13 (leaving out observation 100)

Fitting model 5 out of 13 (leaving out observation 105)

Fitting model 6 out of 13 (leaving out observation 165)

Fitting model 7 out of 13 (leaving out observation 186)

Fitting model 8 out of 13 (leaving out observation 198)

Fitting model 9 out of 13 (leaving out observation 199)

Fitting model 10 out of 13 (leaving out observation 245)

Fitting model 11 out of 13 (leaving out observation 247)

Fitting model 12 out of 13 (leaving out observation 255)

Fitting model 13 out of 13 (leaving out observation 317)
> loo.acer.diff <- loo(acer.diff, save_psis = TRUE, k_threshold = 0.7)
8 problematic observation(s) found.
Model will be refit 8 times.

Fitting model 1 out of 8 (leaving out observation 27)

Fitting model 2 out of 8 (leaving out observation 105)

Fitting model 3 out of 8 (leaving out observation 144)

Fitting model 4 out of 8 (leaving out observation 187)

Fitting model 5 out of 8 (leaving out observation 198)

Fitting model 6 out of 8 (leaving out observation 199)

Fitting model 7 out of 8 (leaving out observation 255)

Fitting model 8 out of 8 (leaving out observation 317)
> loo.acer.mean <- loo(acer.mean, save_psis = TRUE, k_threshold = 0.7)
```

```
10 problematic observation(s) found.
Model will be refit 10 times.

Fitting model 1 out of 10 (leaving out observation 32)

Fitting model 2 out of 10 (leaving out observation 98)

Fitting model 3 out of 10 (leaving out observation 100)

Fitting model 4 out of 10 (leaving out observation 105)

Fitting model 5 out of 10 (leaving out observation 144)

Fitting model 6 out of 10 (leaving out observation 187)

Fitting model 7 out of 10 (leaving out observation 198)

Fitting model 8 out of 10 (leaving out observation 199)

Fitting model 9 out of 10 (leaving out observation 232)

Fitting model 10 out of 10 (leaving out observation 255)
>
> loo_compare(loo.acer.null, loo.acer.day, loo.acer.night, loo.acer.max,
+     loo.acer.min, loo.acer.dawn, loo.acer.dusk, loo.acer.diff, loo.acer.mean)
           elpd_diff se_diff
acer.diff   0.0       0.0
acer.max   -1.3       2.3
acer.dawn  -3.0       2.7
acer.mean  -3.2       3.2
acer.min   -3.5       1.3
acer.dusk  -3.6       2.8
acer.null  -4.1       3.0
acer.day   -5.0       2.5
acer.night -7.1       2.6
>
> # The above works, we don't need the pairwise comparisons ...
> # but leaving these in to show that:
> loo_compare(loo.acer.null, loo.acer.max)
          elpd_diff se_diff
acer.max   0.0       0.0
acer.null -2.8       4.5
> loo_compare(loo.acer.null, loo.acer.diff)
          elpd_diff se_diff
acer.diff  0.0       0.0
acer.null -4.1       3.0
>
> # Now check what seems to matter and run additional models
> # I am looking at estimate versus SD and also the 10-90% intervals
> summary(acer.day, pars="DayT") # marginal

Model Info:
 function:     stan_lmer
 family:       gaussian [identity]
 formula:      BBdelay ~ DayT + (1 | Tree_ID/Cutting_code)
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 319
 groups:       Cutting_code:Tree_ID (116), Tree_ID (10)

Estimates:
       mean   sd   10%   50%   90%
DayT -0.2    0.1 -0.3  -0.2   0.0

MCMC diagnostics
     mcse Rhat n_eff
DayT 0.0  1.0  1545

For each parameter, mcse is Monte Carlo standard error, 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).
> summary(acer.night, pars="NightT") # marginal

Model Info:
 function:     stan_lmer
 family:       gaussian [identity]
 formula:      BBdelay ~ NightT + (1 | Tree_ID/Cutting_code)
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 319
 groups:       Cutting_code:Tree_ID (116), Tree_ID (10)

Estimates:
         mean   sd   10%   50%   90%
NightT 0.1    0.1  0.0   0.1   0.2
```

```
MCMC diagnostics
       mcse Rhat n_eff
NightT 0.0  1.0  2201

For each parameter, mcse is Monte Carlo standard error, 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).
> summary(acer.max, pars="MaxT") # impt

Model Info:
 function:     stan_lmer
 family:       gaussian [identity]
 formula:      BBdelay ~ MaxT + (1 | Tree_ID/Cutting_code)
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 319
 groups:       Cutting_code:Tree_ID (116), Tree_ID (10)

Estimates:
       mean   sd   10%   50%   90%
MaxT -0.8    0.2 -1.0  -0.8  -0.5

MCMC diagnostics
     mcse Rhat n_eff
MaxT 0.0  1.0  1792

For each parameter, mcse is Monte Carlo standard error, 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).
> summary(acer.min, pars="MinT") # impt

Model Info:
 function:     stan_lmer
 family:       gaussian [identity]
 formula:      BBdelay ~ MinT + (1 | Tree_ID/Cutting_code)
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 319
 groups:       Cutting_code:Tree_ID (116), Tree_ID (10)

Estimates:
       mean   sd   10%   50%   90%
MinT 0.5    0.2  0.3   0.5   0.7

MCMC diagnostics
     mcse Rhat n_eff
MinT 0.0  1.0  1428

For each parameter, mcse is Monte Carlo standard error, 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).
> summary(acer.dawn, pars="DawnT") # ns

Model Info:
 function:     stan_lmer
 family:       gaussian [identity]
 formula:      BBdelay ~ DawnT + (1 | Tree_ID/Cutting_code)
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 319
 groups:       Cutting_code:Tree_ID (116), Tree_ID (10)

Estimates:
        mean   sd   10%   50%   90%
DawnT  0.0    0.1 -0.2   0.0   0.1

MCMC diagnostics
      mcse Rhat n_eff
DawnT 0.0  1.0  1521

For each parameter, mcse is Monte Carlo standard error, 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).
> summary(acer.dusk, pars="DuskT") # nothing!

Model Info:
 function:     stan_lmer
 family:       gaussian [identity]
 formula:      BBdelay ~ DuskT + (1 | Tree_ID/Cutting_code)
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 319
 groups:       Cutting_code:Tree_ID (116), Tree_ID (10)

Estimates:
        mean   sd   10%   50%   90%
DuskT  0.0    0.1 -0.1   0.0   0.1
```

```
MCMC diagnostics
      mcse Rhat n_eff
DuskT 0.0  1.0  1695

For each parameter, mcse is Monte Carlo standard error, 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).
> summary(acer.diff, pars="DIFF") # impt

Model Info:
 function:     stan_lmer
 family:       gaussian [identity]
 formula:      BBdelay ~ DIFF + (1 | Tree_ID/Cutting_code)
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 319
 groups:       Cutting_code:Tree_ID (116), Tree_ID (10)

Estimates:
       mean   sd   10%   50%   90%
DIFF -0.3    0.1 -0.5  -0.3  -0.2

MCMC diagnostics
     mcse Rhat n_eff
DIFF 0.0  1.0  1910

For each parameter, mcse is Monte Carlo standard error, 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).
> summary(acer.mean, pars="MeanT") # ns (but loo suggests it's predictively similar to max and min)

Model Info:
 function:     stan_lmer
 family:       gaussian [identity]
 formula:      BBdelay ~ MeanT + (1 | Tree_ID/Cutting_code)
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 319
 groups:       Cutting_code:Tree_ID (116), Tree_ID (10)

Estimates:
        mean   sd   10%   50%   90%
MeanT -0.1    0.5 -0.8  -0.1   0.5

MCMC diagnostics
      mcse Rhat n_eff
MeanT 0.0  1.0  1532

For each parameter, mcse is Monte Carlo standard error, 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).
>
> acer.maxdiff <- stan_lmer(BBdelay ~ MaxT + DIFF + (1|Tree_ID/Cutting_code), data=acer, cores=4)
> acer.mindiff <- stan_lmer(BBdelay ~ MinT + DIFF + (1|Tree_ID/Cutting_code), data=acer, cores=4)
> acer.maxmin <- stan_lmer(BBdelay ~  MaxT + MinT + (1|Tree_ID/Cutting_code), data=acer, cores=4)
>
> loo.acer.maxdiff <- loo(acer.maxdiff, save_psis = TRUE, k_threshold = 0.7)
10 problematic observation(s) found.
Model will be refit 10 times.

Fitting model 1 out of 10 (leaving out observation 98)

Fitting model 2 out of 10 (leaving out observation 100)

Fitting model 3 out of 10 (leaving out observation 105)

Fitting model 4 out of 10 (leaving out observation 198)

Fitting model 5 out of 10 (leaving out observation 199)

Fitting model 6 out of 10 (leaving out observation 237)

Fitting model 7 out of 10 (leaving out observation 241)

Fitting model 8 out of 10 (leaving out observation 247)

Fitting model 9 out of 10 (leaving out observation 255)

Fitting model 10 out of 10 (leaving out observation 317)
> loo.acer.mindiff <- loo(acer.mindiff, save_psis = TRUE, k_threshold = 0.7)
12 problematic observation(s) found.
Model will be refit 12 times.

Fitting model 1 out of 12 (leaving out observation 96)

Fitting model 2 out of 12 (leaving out observation 98)
Fitting model 3 out of 12 (leaving out observation 100)

Fitting model 4 out of 12 (leaving out observation 105)

Fitting model 5 out of 12 (leaving out observation 187)

Fitting model 6 out of 12 (leaving out observation 198)

Fitting model 7 out of 12 (leaving out observation 199)

Fitting model 8 out of 12 (leaving out observation 245)

Fitting model 9 out of 12 (leaving out observation 246)

Fitting model 10 out of 12 (leaving out observation 247)

Fitting model 11 out of 12 (leaving out observation 252)

Fitting model 12 out of 12 (leaving out observation 264)
> loo.acer.maxmin <- loo(acer.maxmin, save_psis = TRUE, k_threshold = 0.7)
9 problematic observation(s) found.
Model will be refit 9 times.

Fitting model 1 out of 9 (leaving out observation 98)

Fitting model 2 out of 9 (leaving out observation 100)

Fitting model 3 out of 9 (leaving out observation 105)

Fitting model 4 out of 9 (leaving out observation 145)

Fitting model 5 out of 9 (leaving out observation 187)

Fitting model 6 out of 9 (leaving out observation 198)

Fitting model 7 out of 9 (leaving out observation 199)

Fitting model 8 out of 9 (leaving out observation 247)

Fitting model 9 out of 9 (leaving out observation 252)
>
> loo_compare(loo.acer.maxdiff, loo.acer.mindiff, loo.acer.maxmin) # these models are all identical
             elpd_diff se_diff
acer.mindiff  0.0       0.0   
acer.maxmin  -0.6       0.8   
acer.maxdiff -0.9       0.9   
>
```