HW8_INLA&otherlibs

Beth Babcock

2024-03-18

Purpose:

The purpose of this markdown document is to work through Homework 8 in Dr. Babcock's Bayesian Statistics Course at the University of Miami. Homework 8 deals with INLA and other libraries in R.

General Start to Code

```
rm(list = ls())

######github#####
#note, only needed after 90 days from 1/16/2024

#  usethis::create_github_token()
#  gitcreds::gitcreds_set()

#####check for r updates#####
#note, updateing may take some time so plan accordingly

#require(installr)

#check.for.updates.R()

#updateR() #only if needed

#######check for package updates#####
#note, updateing may take some time so plan accordingly

#old.packages()

# update.packages() #make the decision to the update the packages
```

Load packages

```
library(INLAutils)
library(INLA)
library(tidyverse)
library(R2jags)
```

```
library(rstan)
library(ggmcmc)
library(purrr)
library(magrittr)
library(here)
library(loo)
library(DHARMa)
library(lme4)
library(rstanarm)
library(shinystan)
library(BayesFactor)

theme_set(theme_bw(base_size=15))
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
```

Data

For this problem we will use the Arabidopsis dataset in the lme4 library. See the help file for the dataset in the lme4 library for an explanation of the variables. The response (y) variable is number of fruits produced (an integer), The fixed effects variables are the factors called nutrients (two different nutrient treatments) and amd (with or without simulated herbivory) and their interaction. Random effects are popu (population) and gen (genotype). Genotype is nested within population, and you can model it as +(1|popu)+(1|gen) in lme4 syntax, because both random effects have a mean of zero in the linear model.

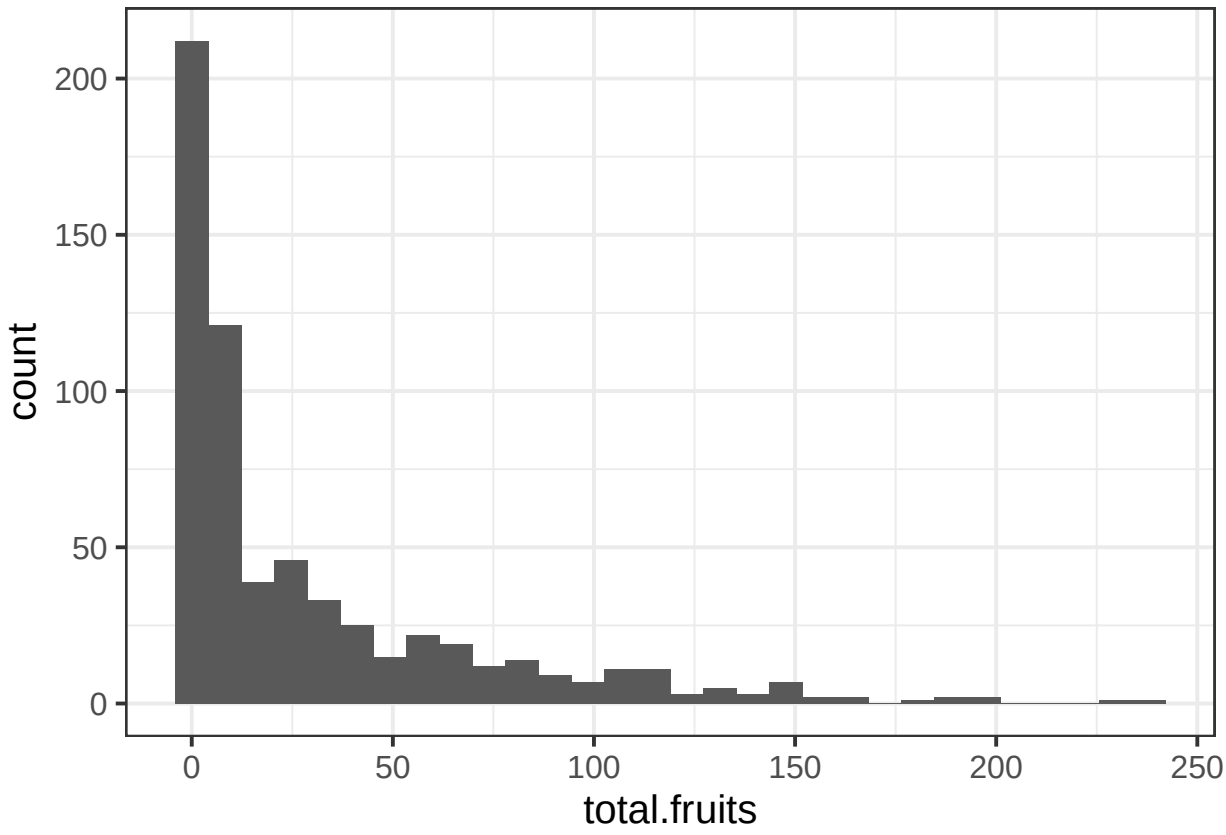
```
#get the data from lme4

arabidopsis.data <- Arabidopsis |>
  mutate(nutrient.factor = as.factor(nutrient),
         amd.factor = as.factor(amd),
         popu.factor = as.factor(popu),
         gen.factor = as.factor(gen))


head(arabidopsis.data)
```

Problem 1) Poisson, negative binomial and random effects in INLA

```
arabidopsis.data |>
  ggplot(aes(x = total.fruits))+
  geom_histogram()
```

A-1) Plot a histogram of the total.fruits variable. Which likelihood would seem to be best for this model based on the data distribution?

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

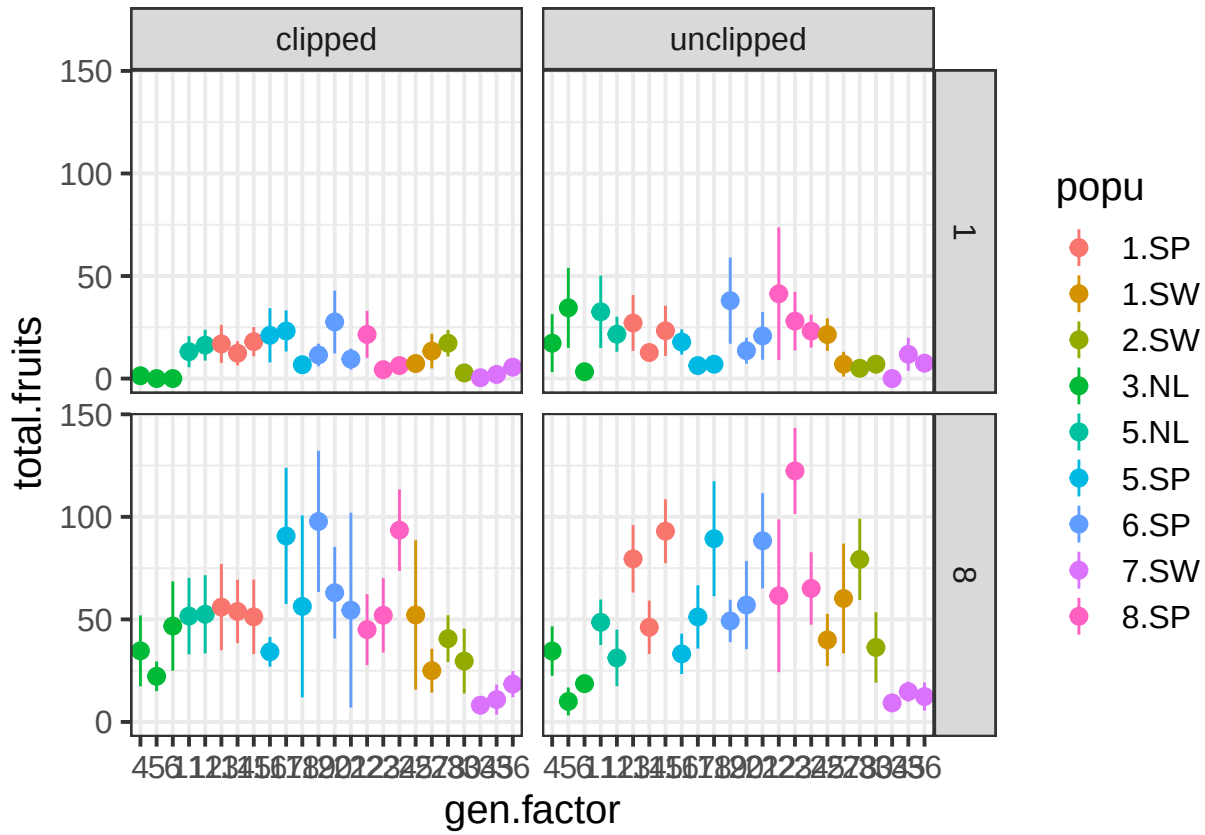


Based on the histogram I would say that a poisson model or negative binomial model would be best for this model. Poisson if the mean and the sd were equal and negative-binomial if the mean and sd are different.

```
arabidopsis.data |>
  ggplot(aes(x = gen.factor, color = popu, y = total.fruits))+
  facet_grid(nutrient.factor~amd.factor)+
  stat_summary()
```

A-2) Use the following code to plot the data. Do you think the random effects are needed and that the fixed effects are likely to be significant?

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```



I think the random effects are needed, and it appears that there is a difference between the nutrients but it is harder to tell if the clipped vs unclipped will be different.

```
#Poission

fruit.poisson.inla <-inla(total.fruits~nutrient.factor +
                                      amd.factor + f(popu.factor,model = "iid") +
                                      f(gen.factor,model = "iid"),
                          data = arabidopsis.data,
                          family = "poisson",
                          control.predictor = list(compute = TRUE),
                          control.compute = list(dic = TRUE,
                                                 waic = TRUE,
                                                 cpo = TRUE))

#Negative Binomial

fruit.negbinom.inla <-inla(total.fruits~nutrient.factor +
                                      amd.factor + f(popu.factor,model = "iid") +
                                      f(gen.factor,model = "iid"),
                          data = arabidopsis.data,
                          family = "nbinomial",
                          control.predictor = list(compute = TRUE),
                          control.compute = list(dic = TRUE,
```

```
                                                waic = TRUE,
                                                cpo = TRUE))


#zero inflated poisson

fruit.zerinflpo.inla <-inla(total.fruits~nutrient.factor +
                                        amd.factor + f(popu.factor,model = "iid") +
                                        f(gen.factor,model = "iid"),
                            data = arabidopsis.data,
                            family = "zeroinflatedpoisson1",
                            control.predictor = list(compute = TRUE),
                            control.compute = list(dic = TRUE,
                                                   waic = TRUE,
                                                   cpo = TRUE))

#gaussian

fruit.gaussian.inla <-inla(total.fruits~nutrient.factor +
                                        amd.factor + f(popu.factor,model = "iid") +
                                        f(gen.factor,model = "iid"),
                            data = arabidopsis.data,
                            control.predictor = list(compute = TRUE),
                            control.compute = list(dic = TRUE,
                                                   waic = TRUE,
                                                   cpo = TRUE))
```

B) Use INLA to run four different likelihood models, each with the same model formula described above, and show the summary statistics. Note that you will change nothing expect the family to do this part. The likelihood models are: 1) Poisson, 2) Negative Binomial, 3) Zero inflated Poisson, and 4) Normal(gaussian).

C) Make a table showing the WAIC and DIC for the 4 likelihood models. Which model is best and is that consistent with the values of the extra hyperparameters for NB, ZIP and Normal models. In other words, is the mean different from the variance and are there extra zeroes? *DIC*

```
DIC.table <- tibble(model = c("Poisson", "Negative Bionomial",
                              "Zero Inflated Poisson", "Gaussian"),
                    dic = c(fruit.poisson.inla$dic$dic,
                            fruit.negbinom.inla$dic$dic,
                            fruit.zerinflpo.inla$dic$dic,
                            fruit.gaussian.inla$dic$dic)) |>
  mutate(deltaDIC = dic - min(dic),
         weight = round(exp(-2*deltaDIC)/sum(exp(-2*deltaDIC)),digits = 5))

DIC.table  |> knitr::kable(caption = "DIC Table")
```

Table 1: DIC Table

model	dic	deltaDIC	weight
Poisson	14004.462	8968.812	0
Negative Bionomial	5035.649	0.000	1
Zero Inflated Poisson	12168.231	7132.581	0
Gaussian	6326.020	1290.371	0

waic

```
WAIC.table <- tibble(model = c("Poisson", "Negative Bionomial",
                               "Zero Inflated Poisson", "Gaussian"),
                     waic = c(fruit.poisson.inla$waic$waic,
                              fruit.negbinom.inla$waic$waic,
                              fruit.zerinflpo.inla$waic$waic,
                              fruit.gaussian.inla$waic$waic)) |>
  mutate(deltaWAIC = waic - min(waic),
         weight = round(exp(-2*deltaWAIC)/sum(exp(-2*deltaWAIC)),digits = 5))

WAIC.table  |> knitr::kable(caption = "WAIC Table")
```

Table 2: WAIC Table

model	waic	deltaWAIC	weight
Poisson	18023.624	12985.958	0
Negative Bionomial	5037.666	0.000	1
Zero Inflated Poisson	15812.805	10775.139	0
Gaussian	6327.997	1290.331	0

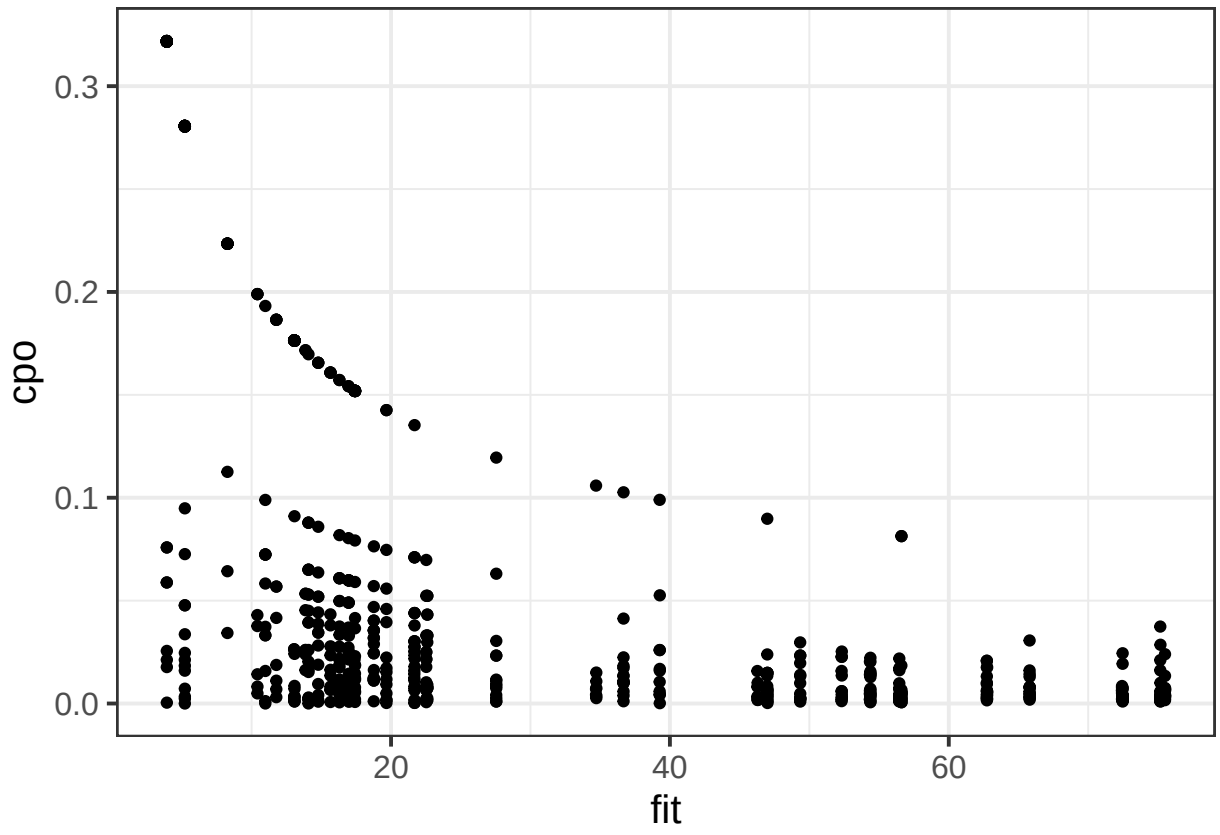
The best model was the negative binomial, indicating that the mean is different from the standard deviation. The zero inflated poisson was not the best model which indicates that there are NOT extra zeros.

D) Plot the PIT and CPO residuals against the predicted values of the data points for the WAIC best model. Does the model seem to fit adequately according to the PIT? *extract residuals*

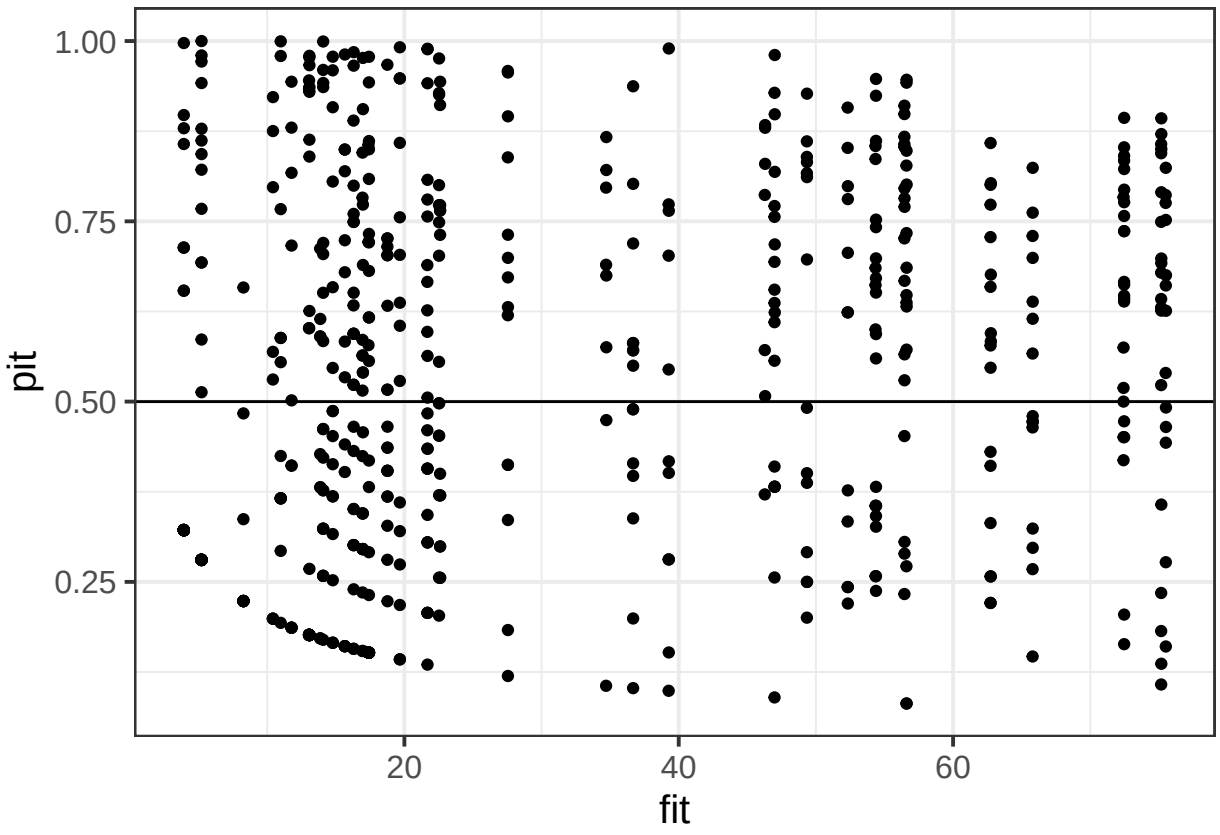
```
fruit.residuals <- arabidopsis.data |>
  mutate(fit = fruit.negbinom.inla$summary.fitted.values$mean,
         cpo = fruit.negbinom.inla$cpo$cpo,
         pit = fruit.negbinom.inla$cpo$pit)
```

residual plots

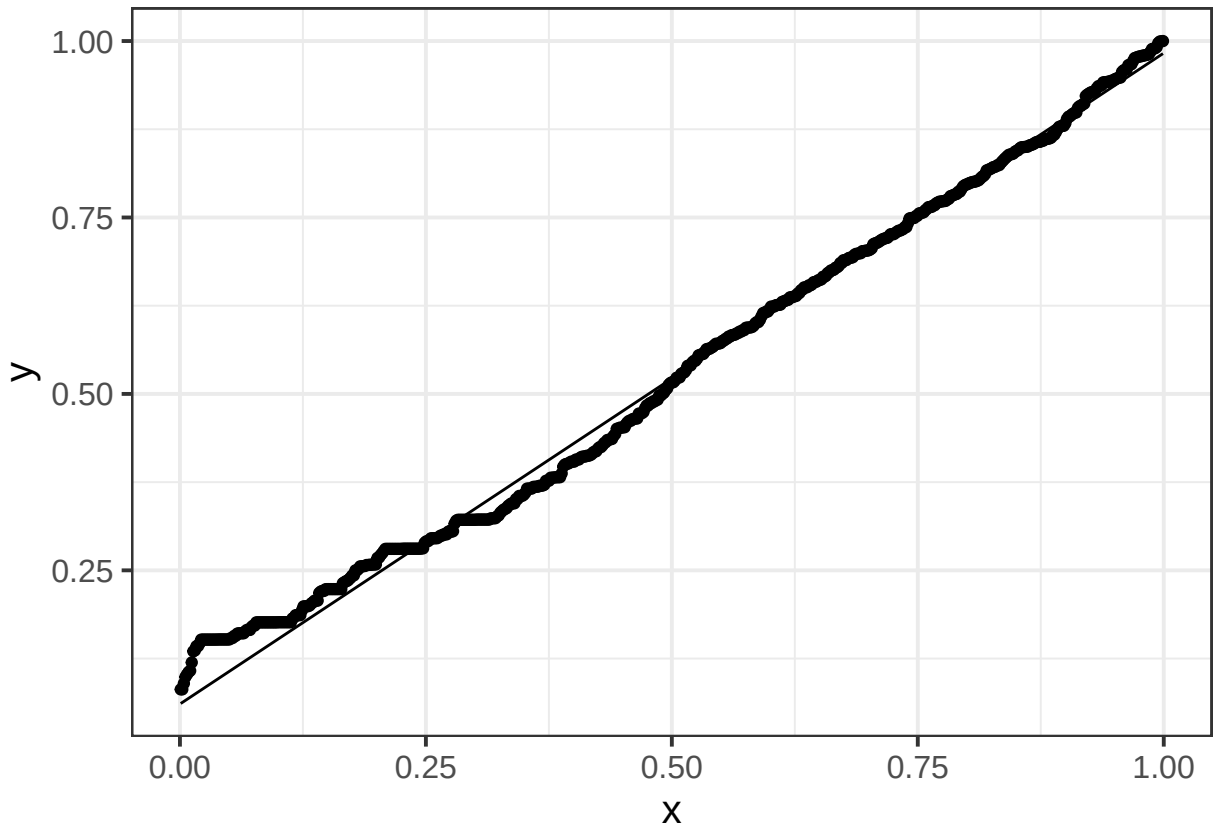
```
fruit.residuals |>
  ggplot(aes(x = fit, y = cpo))+
  geom_point()
```




```
fruit.residuals |>
  ggplot(aes(x = fit, y = pit))+
  geom_point()+
  geom_hline(yintercept = 0.5)
```




```
fruit.residuals |>
  ggplot(aes(sample = pit))+
  geom_qq(distribution = stats::qunif)+
  geom_qq_line(distribution = stats::qunif)
```



The model appears to fit adequately from the PIT.

##Problem 2) Posterior predictive checks and rstanarm

A) Fit the same model from part 1 in rstanarm, using family neg_binomial_2. Do you get similar values of the regression coefficients? *fit and run the model*

```
fruit.negbinom.rstanarm <- stan_glmer(total.fruits ~ nutrient.factor + amd.factor + (1|popu.factor) +
                                                    (1|gen.factor),
                                      data = arabidopsis.data,
                                      family = "neg_binomial_2")
```

results

```
#inla results

summary(fruit.negbinom.inla)
```

```
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts, ", " data = data, quantile
##    quantiles, E = E, offset = offset, ", " scale = scale, weights = weights, Ntrials = Ntrials, strat
##    strata, ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose = verbose, ", " linco
##    lincomb, selection = selection, control.compute = control.compute, ", " control.predictor =
##    control.predictor, control.family = control.family, ", " control.inla = control.inla, control.fixe
```

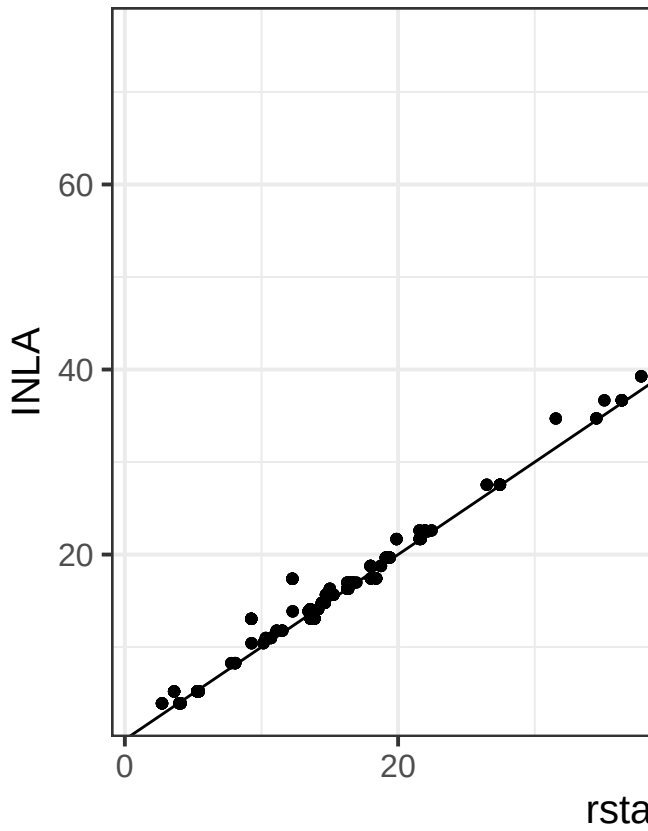
```
##    control.fixed, ", " control.mode = control.mode, control.expert = control.expert, ", " control.haz
##    control.hazard, control.lincomb = control.lincomb, ", " control.update = control.update,
##    control.lp.scale = control.lp.scale, ", " control.pardiso = control.pardiso, only.hyperparam =
##    only.hyperparam, ", " inla.call = inla.call, inla.arg = inla.arg, num.threads = num.threads, ", "
##    keep, working.directory = working.directory, silent = silent, ", " inla.mode = inla.mode, safe = I
##    debug = debug, .parent.frame = .parent.frame)" )
## Time used:
##     Pre = 0.235, Running = 0.402, Post = 0.131, Total = 0.768
## Fixed effects:
##                       mean    sd 0.025quant 0.5quant 0.975quant  mode kld
## (Intercept)          2.435 0.194      2.050    2.435      2.820 2.435   0
## nutrient.factor8     1.206 0.113      0.984    1.206      1.427 1.206   0
## amd.factorunclipped 0.287 0.113      0.064    0.287      0.509 0.287   0
##
## Random effects:
##   Name     Model
##     popu.factor IID model
##    gen.factor IID model
##
## Model hyperparameters:
##
## Deviance Information Criterion (DIC) ...............: 5035.65
## Deviance Information Criterion (DIC, saturated) ....: 759.26
## Effective number of parameters .....................: 10.94
##
## Watanabe-Akaike information criterion (WAIC) ...: 5037.67
## Effective number of parameters .................: 12.03
##
## Marginal log-Likelihood:  -2551.84
## CPO, PIT is computed
## Posterior summaries for the linear predictor and the fitted values are computed
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

```
#rstanarm results

rstanarm_summary <- rownames_to_column(as.data.frame(fruit.negbinom.rstanarm$stan_summary),"parameter")
  filter(parameter %in% c("(Intercept)","nutrient.factor8", "amd.factorunclipped"))

rstanarm_summary
```

The coefficient are similar to the INLA results.

```
fruit.residuals |>
  mutate(rstanarm.fit = fruit.negbinom.rstanarm$fitted.values) |>
  ggplot(aes(x = rstanarm.fit, y = fit))+
  geom_point()+
  geom_abline(intercept = 0, slope = 1)+
  labs(x = "rstanarm", y = "INLA")
```

B) Make a plot of the INLA mean predicted value against the rstan arm mean predictions, be-



ing sure you have them both in the original (not log) scale.

```
#launch_shinystan(fruit.negbinom.rstanarm)
```

C) Look at the diagnostics in shinystan, and print out: 1) the density plot of the posterior predictive distributions and the real data, and 2) the scatterplot of the posterior predictive distribution against the real data. see attached word document for printed plots

Problem 3) Posterior model probabilities

```
arabidopsis.data.bf <- arabidopsis.data |>
  mutate(log_totalfruits = log(total.fruits + 1))

fruit.bf <- anovaBF(log_totalfruits~nutrient.factor*amd.factor, data = arabidopsis.data.bf)
```

A) Using the same dataset, calculate a new Y variable that is log(total.fruits+1) to account for the zero observations. Now use the Bayesfactor library to find the best combination of fixed effect predictor variables. Start with nutrient, treatment and the interaction. Leave out the random effects. Assume Y is normally distributed.

```
##   |
```

```
fruit.bf
```

```
## Bayes factor analysis
## --------------
## [1] nutrient.factor                                          : 1.809334e+19 ±0%
## [2] amd.factor                                               : 1.637316     ±0.01%
## [3] nutrient.factor + amd.factor                             : 6.7801e+19   ±2.03%
## [4] nutrient.factor + amd.factor + nutrient.factor:amd.factor : 1.269602e+19 ±1.28%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
plot(fruit.bf)
```

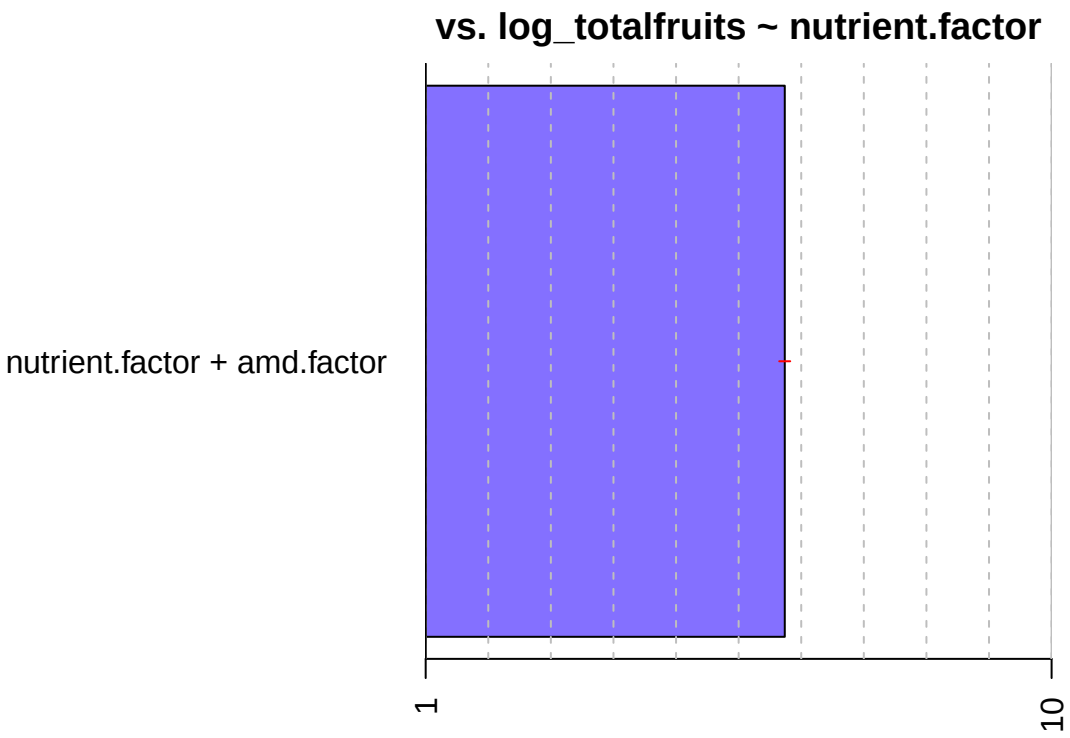

B) Plot the Bayes factors for all models. Which is better?
The additive effects of both nutrient and amd produced the best model.

```
fruit.bf[3]/fruit.bf[1]
```

C) Calculate the odds for the best model vs. the 2nd best model. How much better is the best one?

```
## Bayes factor analysis
## --------------
## [1] nutrient.factor + amd.factor : 3.74729 ±2.03%
##
## Against denominator:
##   log_totalfruits ~ nutrient.factor
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
plot(fruit.bf[3]/fruit.bf[1])
```



The top model is 3.75% better then the second best model (nutrient effect only). Which isn't all that much different.