HW7_anova and INLA

N. Barrus

2024-03-05

Purpose:

The purpose of this markdown document is to work through Homework 7 in Dr. Babcock's Bayesian Statistics Course at the University of Miami. Homework 7 deals with ANOVA and INLA

General Start to Code

```r
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

```r
library(INLAutils)
library(INLA)
library(tidyverse)
library(R2jags)
library(rstan)
library(ggmcmc)
library(purrr)
library(magrittr)
library(here)
library(loo)
library(DHARMa)
theme_set(theme_bw(base_size=15))
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
```

Data

The moth data will be used in the problem 1, and it contains three variables (Moth, Location, Bait). Moth is the number of spruce moths caught in a trap in 48 hours, Location is the location of the trap in the trees (Ground, Lower, Middle, Top), and Bait is the bait used (Chemical, Scent, Sugar).

The football data is the weight in lbs of randomly selected football players from randomly selected teams (coded as 1 to 5).

```
###problem 1 data

moth.data <- read_csv(here("data","moth.csv")) |>
  mutate(LocationNum = as.numeric(as.factor(Location)),
         BaitNum = as.numeric(as.factor(Bait)))
```

```
## Rows: 60 Columns: 3
## -- Column specification ----------------------------------------------------------------------------
## Delimiter: ","
## chr (2): Location, Bait
## dbl (1): Moths
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
head(moth.data)

###problem 2 data

football.data <- read_csv(here("data","football.csv"))
```

```
## Rows: 85 Columns: 2
## -- Column specification ----------------------------------------------------------------------------
## Delimiter: ","
## dbl (2): Team, Weight
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
head(football.data)
```

Problem 1: Fixed effects 2-way ANOVA JAGS

A) Fit an ANOVA model in JAGS to predict number of moths caught (y) as a function of the categorical (x) variables LocationNum and BaitNum. Do not include an interaction. Assume the error is normally distributed and all the factors are fixed effects. Use Chemical and Ground as the reference levels and estimate the differences from that reference level for the other levels of each factor. Use uninformative priors. Give the summary statistics. Which coefficients appear to be different from zero based on their credible intervals? table of contrasts
———————BAIT Period——————————
—————1—————2——————3—————
Location 1 –ref—————ref+Bait[2]—————ref+Bait[3]———
Location 2 –ref+loc[2]——ref+loc[2]+Bait[2]—-ref+Loc[2]+Bait[3]-
Location 3 –ref+loc[3]——ref+loc[3]+Bait[2]—-ref+Loc[3]+Bait[3]-
Location 4 –ref+loc[4]——ref+loc[4]+Bait[2]—-ref+Loc[4]+Bait[3]-

set up and run JAGS model

```
write("model
{  # two-way ANOVA without interaction term, using reference classes

  #set up contrast priors

  base ~ dnorm(0, 1.0E-6)
  a[1] <- 0   #reference location
  a[2] ~ dnorm(0.0, 1.0E-6)
  a[3] ~ dnorm(0.0, 1.0E-6)
  a[4] ~ dnorm(0.0, 1.0E-6)
  b[1] <- 0   #refernce bait
  b[2] ~ dnorm(0.0, 1.0E-6)
  b[3] ~ dnorm(0.0, 1.0E-6)

  tau ~ dgamma(0.001, 0.001)  # uninformative precision

  for (i in 1:N) # for each of the samples
  {
    ymean[i] <- base + a[X1[i]] + b[X2[i]]
    Y[i] ~ dnorm(ymean[i], tau)
    LL[i] <- -0.5*log(2*3.14159)+0.5*log(tau)-0.5*tau*(Y[i]-ymean[i])*(Y[i]-ymean[i])
  }
}
",file=here("JAGS_mods","HW7-1A-ANOVA-2way-noint.txt"))

moth.list <- list(Y = moth.data$Moths,
                  X1 = moth.data$LocationNum,
                  X2 = moth.data$BaitNum,
                  N = length(moth.data$Moths))

jags.anova.2way.noint <-jags(data = moth.list,
  model.file=here("JAGS_mods","HW7-1A-ANOVA-2way-noint.txt"),
  parameters.to.save=c("base","a","b","tau","LL"),n.chains=2,n.iter=110000,n.burnin=10000,n.thin=10)
```

output summary

```r
LLs <- paste0("LL[",1:moth.list$N,"]")

summary.table <- as.data.frame(round(jags.anova.2way.noint$BUGSoutput$summary,digits = 3)) |>
  rownames_to_column() |>
  as.tibble() |>
  filter(!(rowname %in% LLs))|>
  rename(parameter = rowname)

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

Table 1: Summary Table

parameter	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
a[1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1
a[2]	14.289	2.843	8.728	12.408	14.285	16.167	19.867	1.001	20000
a[3]	11.934	2.862	6.355	10.027	11.926	13.841	17.630	1.001	20000
a[4]	4.248	2.853	-1.342	2.329	4.268	6.155	9.924	1.001	15000
b[1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1
b[2]	-2.766	2.467	-7.655	-4.417	-2.757	-1.127	2.060	1.001	20000
b[3]	0.295	2.465	-4.591	-1.357	0.309	1.957	5.156	1.001	20000
base	19.890	2.479	15.041	18.242	19.869	21.558	24.784	1.001	20000
deviance	415.342	3.940	409.665	412.407	414.666	417.527	424.691	1.001	20000
tau	0.017	0.003	0.011	0.015	0.017	0.019	0.024	1.001	20000

The reference class (Ground, Chemical), the location Lower (i.e., a[2]), the location Middle (i.e., a[3]).

B) Now add the 2-way interaction between location and bait to your model. Give the summary statistics. Which coefficients appear to be different from zero based on their credible intervals?
table of contrasts

——————————————BAIT Period——————————————

——————1———————-2———————————-3—————

Location 1 –ref——————————ref+Bait[2]——————ref+Bait[3]————

Location 2 –ref+loc[2]——ref+loc[2]+Bait[2]+int[2,2]—-ref+Loc[2]+Bait[3]+int[2,3]-

Location 3 –ref+loc[3]——ref+loc[3]+Bait[2]+int[3,2]—-ref+Loc[3]+Bait[3]+int[3,3]-

Location 4 –ref+loc[4]——ref+loc[4]+Bait[2]+int[4,2]—-ref+Loc[4]+Bait[3]+int[4,3]-

set up and run JAGS model

```r
write("model
{ #two-way ANOVA with interaction term, using reference classes

  #set up contrast priors- noninteractions
  base ~ dnorm(0, 1.0E-6)
  a[1] <- 0   #reference location
  a[2] ~ dnorm(0.0, 1.0E-6)
  a[3] ~ dnorm(0.0, 1.0E-6)
  a[4] ~ dnorm(0.0, 1.0E-6)
  b[1] <- 0   #reference bait
  b[2] ~ dnorm(0.0, 1.0E-6)
  b[3] ~ dnorm(0.0, 1.0E-6)
  #set up contrast priors- interactions
  int[1,1] <- 0
  int[2,1] <- 0
  int[3,1] <- 0
  int[4,1] <- 0
  int[1,2] <- 0
  int[2,2] ~ dnorm(0.0, 1.0E-6)
  int[3,2] ~ dnorm(0.0, 1.0E-6)
  int[4,2] ~ dnorm(0.0, 1.0E-6)
  int[1,3] <- 0
  int[2,3] ~ dnorm(0.0, 1.0E-6)
  int[3,3] ~ dnorm(0.0, 1.0E-6)
  int[4,3] ~ dnorm(0.0, 1.0E-6)

  tau ~ dgamma(0.001, 0.001)  # uninformative precision

#model

  for (i in 1:N) # for each of the samples
  {
    ymean[i] <- base + a[X1[i]] + b[X2[i]] + int[X1[i],X2[i]]
    Y[i] ~ dnorm(ymean[i], tau)
    LL[i] <- -0.5*log(2*3.14159)+0.5*log(tau)-0.5*tau*(Y[i]-ymean[i])*(Y[i]-ymean[i])
  }
}
",file=here("JAGS_mods","HW7-1B-ANOVA-2way-int.txt"))

moth.list <- list(Y = moth.data$Moths,
                  X1 = moth.data$LocationNum,
                  X2 = moth.data$BaitNum,
                  N = length(moth.data$Moths))

jags.anova.2way.int <-jags(data = moth.list,
  model.file=here("JAGS_mods","HW7-1B-ANOVA-2way-int.txt"),
  parameters.to.save=c("base","a","b","int","tau","LL"),n.chains=2,n.iter=110000,n.burnin=10000,n.thin=
```

output summary

```
LLs <- paste0("LL[",1:moth.list$N,"]")

summary.table <- as.data.frame(round(jags.anova.2way.int$BUGSoutput$summary,digits = 3)) |>
  rownames_to_column() |>
  as.tibble() |>
  filter(!(rowname %in% LLs)) |>
  rename(parameter = rowname)

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

Table 2: Summary Table

parameter	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
a[1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1
a[2]	16.776	5.181	6.715	13.336	16.716	20.184	27.037	1.001	20000
a[3]	12.573	5.160	2.593	9.094	12.540	16.024	22.792	1.001	16000
a[4]	3.808	5.151	-6.304	0.357	3.825	7.244	13.967	1.001	20000
b[1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1
b[2]	-2.227	5.197	-12.423	-5.686	-2.227	1.242	8.065	1.001	20000
b[3]	1.773	5.152	-8.336	-1.658	1.745	5.226	11.961	1.001	20000
base	19.200	3.644	12.062	16.747	19.231	21.624	26.328	1.001	20000
deviance	420.336	5.757	411.371	416.136	419.559	423.712	433.504	1.001	20000
int[1,1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1
int[2,1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1
int[3,1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1
int[4,1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1
int[1,2]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1
int[2,2]	-0.948	7.342	-15.392	-5.894	-0.929	3.948	13.383	1.001	20000
int[3,2]	-1.681	7.324	-16.066	-6.592	-1.679	3.200	12.719	1.001	20000
int[4,2]	0.609	7.354	-13.817	-4.296	0.592	5.484	15.112	1.001	20000
int[1,3]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1
int[2,3]	-6.537	7.268	-20.970	-11.318	-6.509	-1.733	7.793	1.001	20000
int[3,3]	-0.150	7.269	-14.439	-5.114	-0.130	4.759	14.169	1.001	20000
int[4,3]	0.848	7.251	-13.351	-3.904	0.841	5.561	15.230	1.001	20000
tau	0.016	0.003	0.010	0.014	0.016	0.018	0.023	1.001	17000

the reference class-Ground, Chemical(i.e, base), the location lower (i.e, a[2]), the location middle (i.e., a[3])

```r
LLs <- paste0("LL[",1:moth.list$N,"]")

WAIC.table <- tibble(model = c("2way-Anova-noint", "2way-Anova-int"),
                    model.ls = c(list(jags.anova.2way.noint), list(jags.anova.2way.int)),
                    LL.ls = c(list(jags.anova.2way.noint$BUGSoutput$sims.matrix[,LLs]),
                            list(jags.anova.2way.int$BUGSoutput$sims.matrix[,LLs]))) |>
  mutate(WAIC.ls = map(LL.ls, waic),
         elpd_waic = map_dbl(WAIC.ls,c(1,1)),
         p_waic = map_dbl(WAIC.ls,c(1,2)),
         waic = map_dbl(WAIC.ls,c(1,3)),
         deltaWAIC = waic - min(waic),
         weight = round(exp(-2*deltaWAIC)/sum(exp(-2*deltaWAIC)),digits = 5))
```

C) Calculate the delta-WAIC. Which of these two models is best? Is that consistent with what you would expect from looking at the credible intervals of the parameter estimates?

```
## Warning: There were 2 warnings in ‘mutate()‘.
## The first warning was:
## i In argument: ‘WAIC.ls = map(LL.ls, waic)‘.
## Caused by warning:
## !
## 2 (3.3%) p_waic estimates greater than 0.4. We recommend trying loo instead.
## i Run ‘dplyr::last_dplyr_warnings()‘ to see the 1 remaining warning.
```

```
WAIC.table |> select(-model.ls,-LL.ls,-WAIC.ls) |> knitr::kable(caption = "WAIC Table")
```

Table 3: WAIC Table

model	elpd_waic	p_waic	waic	deltaWAIC	weight
2way-Anova-noint	-211.1740	6.369937	422.3480	0.00000	1
2way-Anova-int	-216.7617	11.329449	433.5233	11.17532	0

Yes this is consistent with what I would expect, because all of the interaction coefficients CIs included zero (i.e., non-significant).

Problem 2: Mixed effects/ heirarchical models in JAGS or STAN

A) Treating team as a random effect, estimate the mean for each team, the grand mean, the standard deviation between teams and the standard deviation within teams (i.e. error sd). Assume the data are normally distributed. Assume the within group variance is the same for all groups. Use an exponential prior for the standard deviations. Show the summary statistics for all the parameters and use ggs_caterpillar to compare the means across teams. *write and run the model*

```
write("model {
      ##random effects for one-way ANOVA##

      #set up priors
      a ~ dnorm(0,1.0E-6)
      for(i in 1:n.team) {
        d[i] ~ dnorm(0.0, tau_a)
      }
      sd_among ~ dunif(0,100)
      sd_within ~ dunif(0,100)

      #convert sd to precision
      tau_a <- 1/(sd_among*sd_among)
      tau_w <- 1/(sd_within*sd_within)

      #model
      for(i in 1:N){
        ymean[i] <- a +d[team[i]]
        Y[i] ~ dnorm(ymean[i], tau_w)
        LL[i] <- -0.5*log(2*3.14159)+0.5*log(tau_w)-0.5*tau_w*(Y[i]-ymean[i])*(Y[i]-ymean[i])
      }
      }", file = here("JAGS_mods","JAGS_HW7-2A-ME.txt"))

football.list <- list(Y = football.data$Weight,
                      team = football.data$Team,
                      n.team = length(unique(football.data$Team)),
                      N = length(football.data$Weight))
```

```
JAGS.2A.1way.anova.me <- jags(data = football.list,model.file=here("JAGS_mods","JAGS_HW7-2A-ME.txt"),
  parameters.to.save=c("a","d","sd_among","sd_within","LL"),
  n.chains=2,n.iter=110000,n.burnin=10000,n.thin=10)
```

summary output

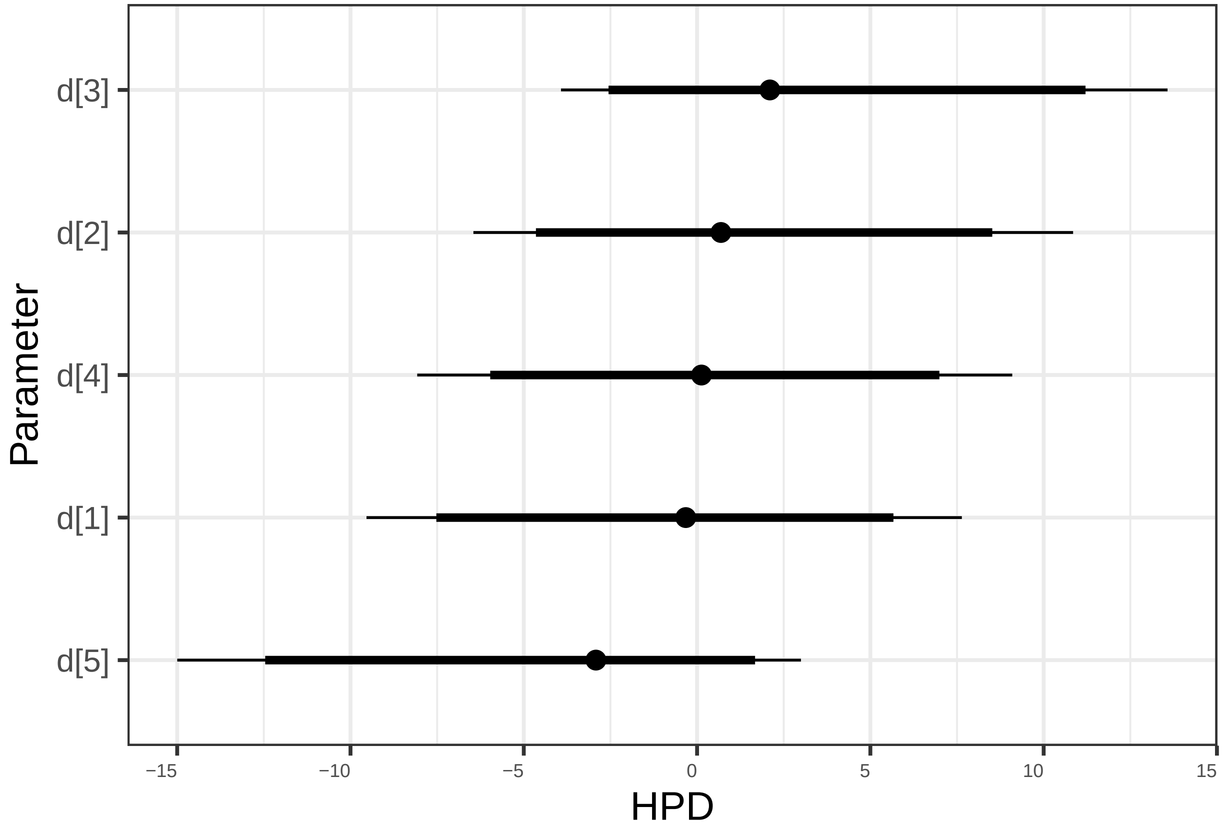
```
LLs <- paste0("LL[",1:football.list$N,"]")

summary.table <- as.data.frame(round(JAGS.2A.1way.anova.me$BUGSoutput$summary,digits = 3)) |>
  rownames_to_column() |>
  as.tibble() |>
  filter(!(rowname %in% LLs ))|>
  rename(parameter = rowname)

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

Table 4: Summary Table

parameter	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
a	248.432	3.776	241.026	246.553	248.438	250.335	255.804	1.001	3300
d[1]	-0.643	4.206	-9.538	-2.510	-0.325	1.229	7.640	1.001	20000
d[2]	1.280	4.238	-6.454	-0.742	0.688	3.153	10.850	1.001	13000
d[3]	2.978	4.562	-3.927	0.094	2.101	5.228	13.575	1.001	20000
d[4]	0.340	4.208	-8.075	-1.515	0.126	2.172	9.094	1.001	6500
d[5]	-3.805	4.783	-14.999	-6.330	-2.918	-0.447	2.997	1.001	9200
deviance	718.926	3.189	714.027	716.587	718.664	720.702	726.339	1.002	2600
sd_among	5.533	5.073	0.254	2.248	4.332	7.203	18.566	1.001	5500
sd_within	16.771	1.343	14.390	15.837	16.680	17.611	19.645	1.001	9800

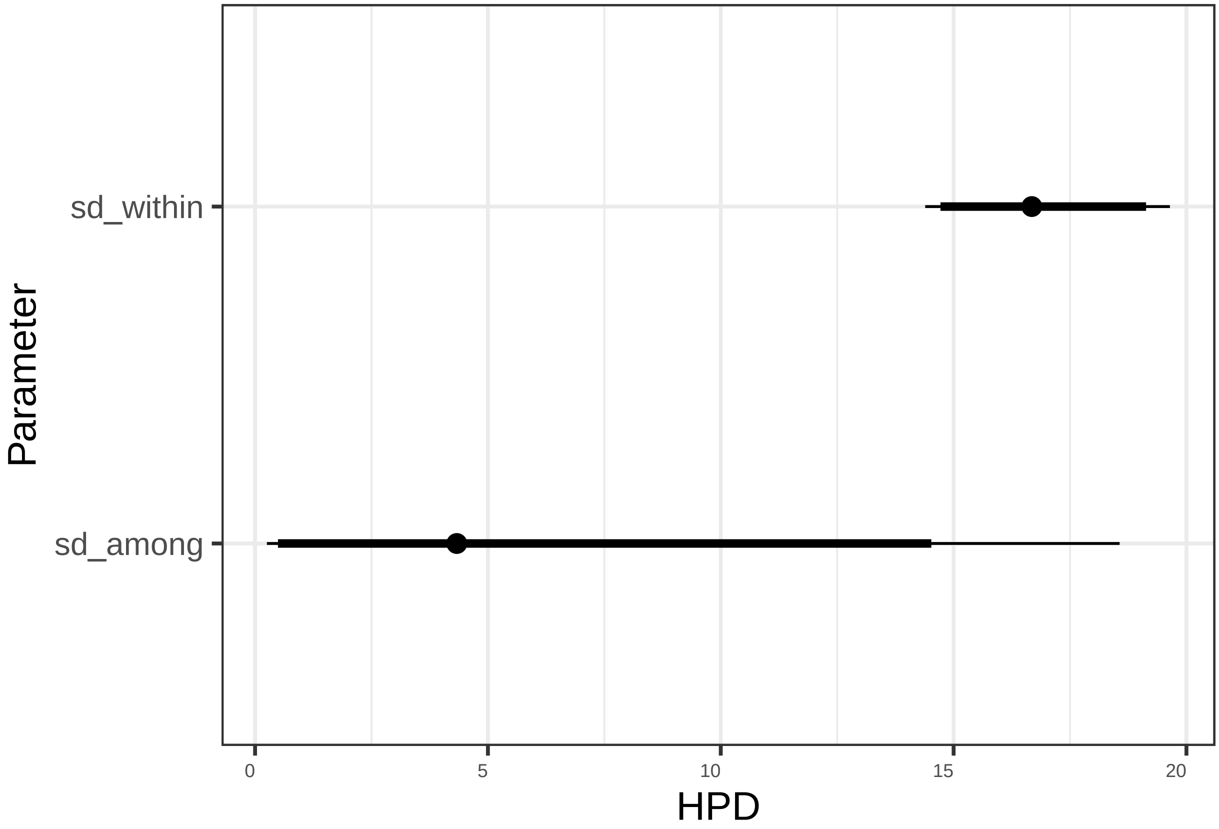
caterpillar plot

```
ggs(as.mcmc(JAGS.2A.1way.anova.me)) |>
  filter(Parameter %in% paste0("d[",1:football.list$n.team,"]")) |>
  ggs_caterpillar()
```



```
ggs(as.mcmc(JAGS.2A.1way.anova.me)) |>
  filter(Parameter %in% c("sd_within","sd_among")) |>
  ggs_caterpillar()
```

B) Compare the CIs of the standard deviations within and between groups. Based on these results, do you think the location random effect is necessary in this model?



Based on these results, the sd_within is greater than sd_among, thus I would say that random effects aren't necessary in the model.

C) Now run the model without the location random effect, with only the grand mean (i.e a null model). Show the WAIC values for the models in part a and part c. Does the WAIC support including the random effect in the model? *write and run the model*

```
write("model {
      ##Null model ##

      #set up priors
      a ~ dnorm(0,1.0E-6)
      sd_within ~ dunif(0,100)

      #convert sd to precision
      tau_w <- 1/(sd_within*sd_within)

      #model
      for(i in 1:N){
        ymean[i] <- a
        Y[i] ~ dnorm(ymean[i], tau_w)
        LL[i] <- -0.5*log(2*3.14159)+0.5*log(tau_w)-0.5*tau_w*(Y[i]-ymean[i])*(Y[i]-ymean[i])
      }
      }", file = here("JAGS_mods","JAGS_HW7-2A-ME.txt"))

football.list <- list(Y = football.data$Weight,
                      team = football.data$Team,
                      n.team = length(unique(football.data$Team)),
                      N = length(football.data$Weight))

JAGS.2C.1way.anova.null <- jags(data = football.list,model.file=here("JAGS_mods","JAGS_HW7-2A-ME.txt"),
  parameters.to.save=c("a","sd_within","LL"),
  n.chains=2,n.iter=110000,n.burnin=10000,n.thin=10)
```

```
## Warning in jags.model(model.file, data = data, inits = init.values, n.chains = n.chains, : Unused va
## data

## Warning in jags.model(model.file, data = data, inits = init.values, n.chains = n.chains, : Unused va
## data
```

summary output

```
LLs <- paste0("LL[",1:football.list$N,"]")

summary.table <- as.data.frame(round(JAGS.2C.1way.anova.null$BUGSoutput$summary, digits = 3)) |>
  rownames_to_column() |>
  as.tibble() |>
  filter(!(rowname %in% LLs ))|>
  rename(parameter = rowname)

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

Table 5: Summary Table

parameter	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
a	248.481	1.842	244.807	247.257	248.479	249.710	252.131	1.001	20000
deviance	721.071	2.089	719.063	719.603	720.404	721.847	726.748	1.001	15000
sd_within	16.975	1.348	14.605	16.028	16.886	17.821	19.870	1.001	5500

WAIC table

```
LLs <- paste0("LL[",1:football.list$N,"]")

WAIC.table <- tibble(model = c("team_random", "null"),
                    model.ls = c(list(JAGS.2A.1way.anova.me), list(JAGS.2C.1way.anova.null)),
                    LL.ls = c(list(JAGS.2A.1way.anova.me$BUGSoutput$sims.matrix[,LLs]),
                            list(JAGS.2C.1way.anova.null$BUGSoutput$sims.matrix[,LLs]))) |>
  mutate(WAIC.ls = map(LL.ls, waic),
         elpd_waic = map_dbl(WAIC.ls,c(1,1)),
         p_waic = map_dbl(WAIC.ls,c(1,2)),
         waic = map_dbl(WAIC.ls,c(1,3)),
         deltaWAIC = waic - min(waic),
         weight = round(exp(-2*deltaWAIC)/sum(exp(-2*deltaWAIC)),digits = 5))

WAIC.table |> select(-model.ls,-LL.ls,-WAIC.ls) |> knitr::kable(caption = "WAIC Table")
```

Table 6: WAIC Table

model	elpd_waic	p_waic	waic	deltaWAIC	weight
team_random	-361.6429	4.147333	723.2858	0.4411404	0.29271
null	-361.4223	1.726465	722.8447	0.0000000	0.70729

The deltaWAIC of the random effect model was only 0.468 from the null model, suggesting that both models perform similarly. This also suggest that including the random effects doesn't add anything to the model, so i'd say it doesn't support using the random effect in the model.

D) When, if ever, would it make sense to model team as a fixed effect rather than a random effect? Explain your reasoning. It would make sense to model the team as a fixed effect if the goal was to assess different weights between teams. For example, if the research question was: which team has the heaviest players? In this instance, the mean value for each team is of interest.

Problem 3: INLA

A) Repeat the moth models from part 1 in INLA along with a 3rd model with location only, using default priors, and print the results with summary and autoplot. Do you get similar coefficients to part 1 for the same models? *models*

```
#bait+location

moth.noint.inla <- inla(Moths~Location+Bait, data = moth.data,
                        control.compute = list(dic = TRUE,
                                               waic = TRUE))

#bait*location

moth.int.inla <- inla(Moths~Location*Bait, data = moth.data,
                      control.compute = list(dic = TRUE,
                                             waic = TRUE))

#location
moth.loc.inla <- inla(Moths~Location, data = moth.data,
                      control.compute = list(dic = TRUE,
                                             waic = TRUE))
```

model summaries

```
#bait+location

summary(moth.noint.inla)
```

```
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts, ", " data = data, quantil
##    quantiles, E = E, offset = offset, ", " scale = scale, weights = weights, Ntrials = Ntrials, stra
##    strata, ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose = verbose, ", " linc
##    lincomb, selection = selection, control.compute = control.compute, ", " control.predictor =
##    control.predictor, control.family = control.family, ", " control.inla = control.inla, control.fix
##    control.fixed, ", " control.mode = control.mode, control.expert = control.expert, ", " control.ha
##    control.hazard, control.lincomb = control.lincomb, ", " control.update = control.update, control.
##    control.lp.scale, ", " control.pardiso = control.pardiso, only.hyperparam = only.hyperparam, ", "
##    = inla.call, inla.arg = inla.arg, num.threads = num.threads, ", " keep = keep, working.directory
##    working.directory, silent = silent, ", " inla.mode = inla.mode, safe = FALSE, debug = debug, .par
##    = .parent.frame)" )
## Time used:
##     Pre = 0.19, Running = 0.18, Post = 0.447, Total = 0.817
## Fixed effects:
##                  mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept)    19.992 2.393     15.287   19.990     24.708 19.990   0
## LocationLower  14.096 2.762      8.652   14.099     19.524 14.099   0
## LocationMiddle 11.772 2.762      6.328   11.775     17.200 11.775   0
## LocationTop     4.135 2.762     -1.308    4.137      9.564  4.137   0
## BaitScent      -2.735 2.397     -7.453   -2.735      1.984 -2.735   0
## BaitSugar       0.306 2.397     -4.412    0.306      5.025  0.306   0
##
## Model hyperparameters:
##
## Deviance Information Criterion (DIC) ...............: 422.07
## Deviance Information Criterion (DIC, saturated) ....: 69.39
## Effective number of parameters .....................: 7.02
##
## Watanabe-Akaike information criterion (WAIC) ...: 422.32
## Effective number of parameters .................: 6.61
##
## Marginal log-Likelihood:  -231.02
##  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)')
```

```
#bait*location

summary(moth.int.inla)
```

```
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts, ", " data = data, quantil
##    quantiles, E = E, offset = offset, ", " scale = scale, weights = weights, Ntrials = Ntrials, stra
##    strata, ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose = verbose, ", " linc
##    lincomb, selection = selection, control.compute = control.compute, ", " control.predictor =
##    control.predictor, control.family = control.family, ", " control.inla = control.inla, control.fix
##    control.fixed, ", " control.mode = control.mode, control.expert = control.expert, ", " control.ha
##    control.hazard, control.lincomb = control.lincomb, ", " control.update = control.update, control.
##    control.lp.scale, ", " control.pardiso = control.pardiso, only.hyperparam = only.hyperparam, ", "
##    = inla.call, inla.arg = inla.arg, num.threads = num.threads, ", " keep = keep, working.directory
##    working.directory, silent = silent, ", " inla.mode = inla.mode, safe = FALSE, debug = debug, .par
##    = .parent.frame)" )
## Time used:
##     Pre = 0.201, Running = 0.144, Post = 0.92, Total = 1.26
## Fixed effects:
##                            mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept)              19.630 3.318     13.120   19.623     26.177 19.624   0
## LocationLower            16.095 4.717      6.783   16.106     25.343 16.106   0
## LocationMiddle           12.012 4.716      2.707   12.021     21.264 12.020   0
## LocationTop               3.357 4.716     -5.939    3.364     12.616  3.364   0
## BaitScent                -2.603 4.674    -11.815   -2.598      6.576 -2.598   0
## BaitSugar                 1.305 4.675     -7.912    1.312     10.480  1.312   0
## LocationLower:BaitScent  -0.318 6.656    -13.383   -0.328     12.805 -0.328   0
## LocationMiddle:BaitScent -1.223 6.655    -14.293   -1.232     11.894 -1.231   0
## LocationTop:BaitScent     1.003 6.655    -12.074    0.998     14.111  0.998   0
## LocationLower:BaitSugar  -5.758 6.656    -18.817   -5.771      7.374 -5.771   0
## LocationMiddle:BaitSugar  0.448 6.656    -12.619    0.438     13.569  0.438   0
## LocationTop:BaitSugar     1.292 6.655    -11.782    1.285     14.404  1.285   0
##
## Model hyperparameters:
##
## Deviance Information Criterion (DIC) ...............: 432.04
## Deviance Information Criterion (DIC, saturated) ....: 75.11
## Effective number of parameters .....................: 12.74
##
## Watanabe-Akaike information criterion (WAIC) ...: 432.81
## Effective number of parameters .................: 11.62
##
## Marginal log-Likelihood:  -240.54
##  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)’)
```

```
#location

summary(moth.loc.inla)
```

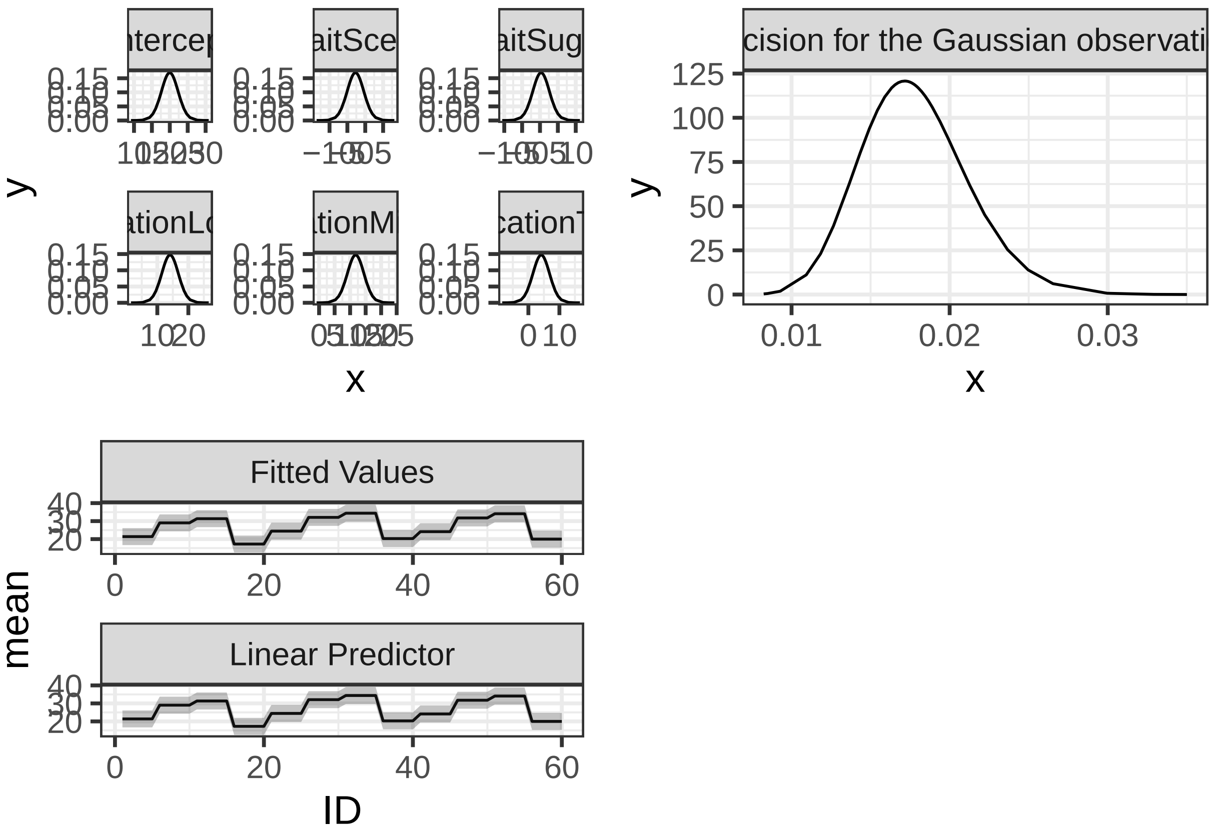
```
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts, ", " data = data, quantil
##    quantiles, E = E, offset = offset, ", " scale = scale, weights = weights, Ntrials = Ntrials, stra
##    strata, ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose = verbose, ", " linc
##    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.ha
##    control.hazard, control.lincomb = control.lincomb, ", " control.update = control.update, control.l
##    control.lp.scale, ", " control.pardiso = control.pardiso, only.hyperparam = only.hyperparam, ", "
##    = inla.call, inla.arg = inla.arg, num.threads = num.threads, ", " keep = keep, working.directory =
##    working.directory, silent = silent, ", " inla.mode = inla.mode, safe = FALSE, debug = debug, .pare
##    = .parent.frame)" )
## Time used:
##     Pre = 0.175, Running = 0.16, Post = 0.686, Total = 1.02
## Fixed effects:
##                  mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept)    19.182 1.953     15.346   19.180     23.030 19.180   0
## LocationLower  14.097 2.762      8.654   14.099     19.522 14.099   0
## LocationMiddle 11.772 2.762      6.330   11.775     17.198 11.775   0
## LocationTop     4.135 2.761     -1.305    4.137      9.562  4.137   0
##
## Model hyperparameters:
##
```

```
## Deviance Information Criterion (DIC) ...............: 420.06
## Deviance Information Criterion (DIC, saturated) ....: 67.38
## Effective number of parameters .....................: 5.00
##
## Watanabe-Akaike information criterion (WAIC) ...: 420.13
## Effective number of parameters .................: 4.73
##
## Marginal log-Likelihood:  -226.71
##  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)')
```

auto plots

```
#bait+location

autoplot(moth.noint.inla)
```

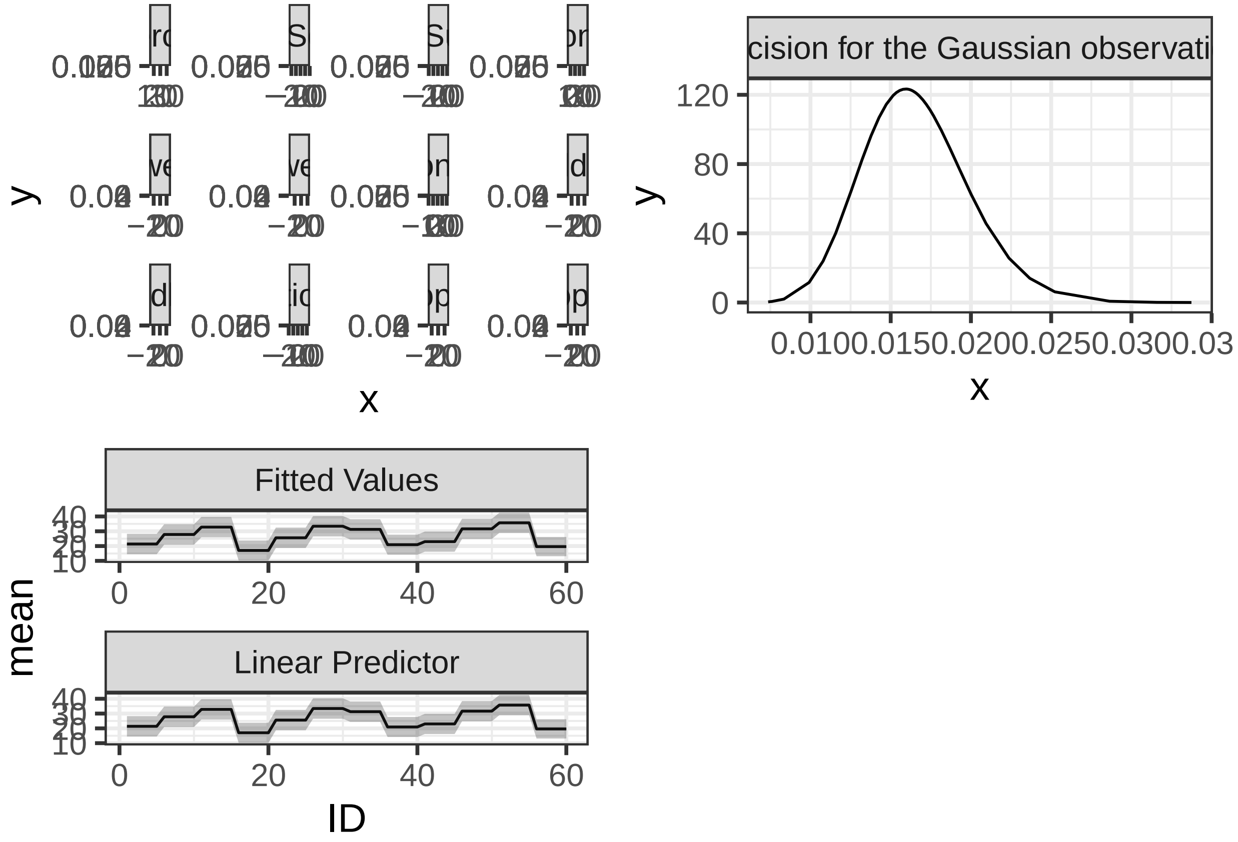
```
## Warning in autoplot.inla(moth.noint.inla): Plot 3 selected in which, but no random effects to plot ma

## Warning in autoplot.inla(moth.noint.inla): Plot 3 will not be plotted.
```



```
#bait*location

autoplot(moth.int.inla)
```

```
## Warning in autoplot.inla(moth.int.inla): Plot 3 selected in which, but no random effects to plot marg
```

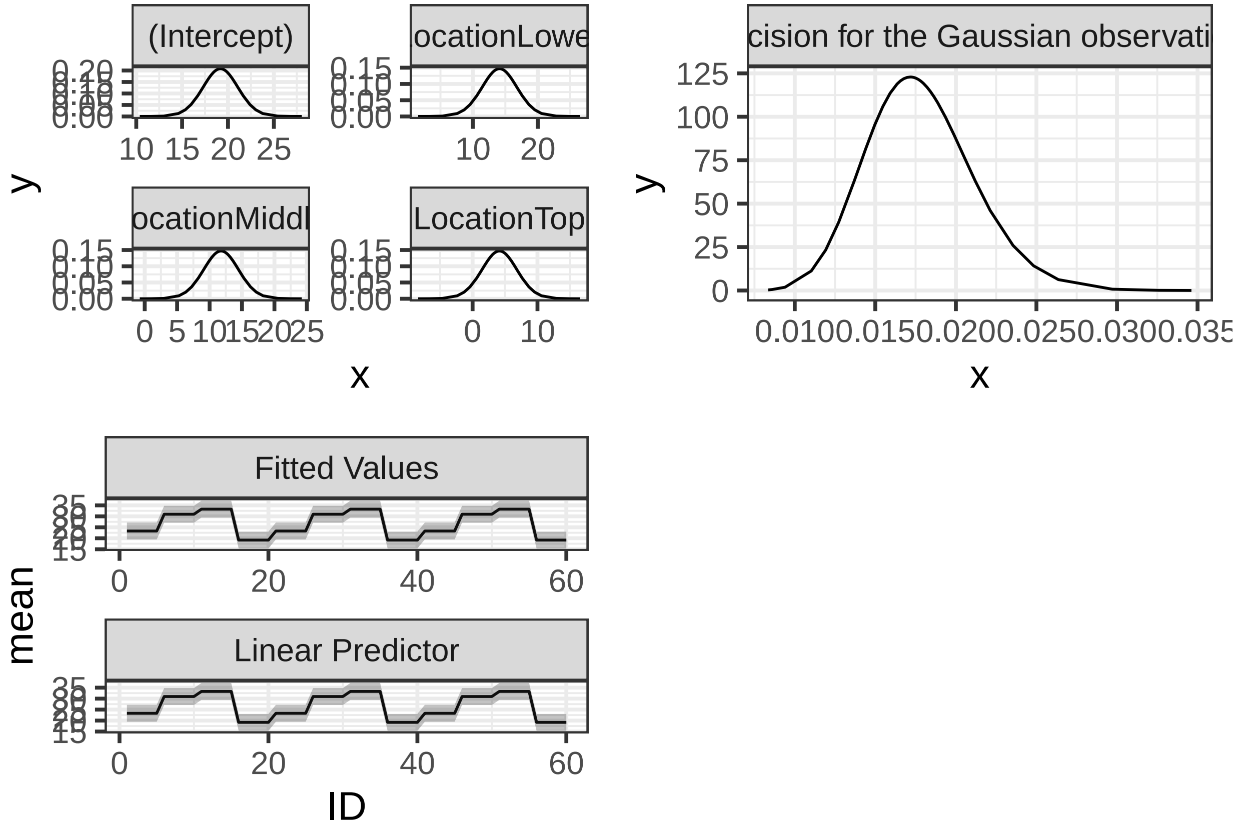
```
## Warning in autoplot.inla(moth.int.inla): Plot 3 will not be plotted.
```



```r
#location
autoplot(moth.loc.inla)
```

```
## Warning in autoplot.inla(moth.loc.inla): Plot 3 selected in which, but no random effects to plot marg
```

```
## Warning in autoplot.inla(moth.loc.inla): Plot 3 will not be plotted.
```



The coefficients were similar in INLA as they were to the JAGS versions in problem 1.

```
###DIC
DIC.table <- tibble(model = c("no_interaction", "interaction","location_only"),
                    dic = c(moth.noint.inla$dic$dic,
                            moth.int.inla$dic$dic,
                            moth.loc.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")
```

B) Calculate DIC and WAIC for all 3 models. Which model is preferred?

Table 7: DIC Table

model	dic	deltaDIC	weight
no_interaction	422.0685	2.010931	0.0176
interaction	432.0433	11.985740	0.0000
location_only	420.0575	0.000000	0.9824

```r
###WAIC

WAIC.table <- tibble(model = c("no_interaction", "interaction","location_only"),
                    waic = c(moth.noint.inla$waic$waic,
                             moth.int.inla$waic$waic,
                             moth.loc.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")
```

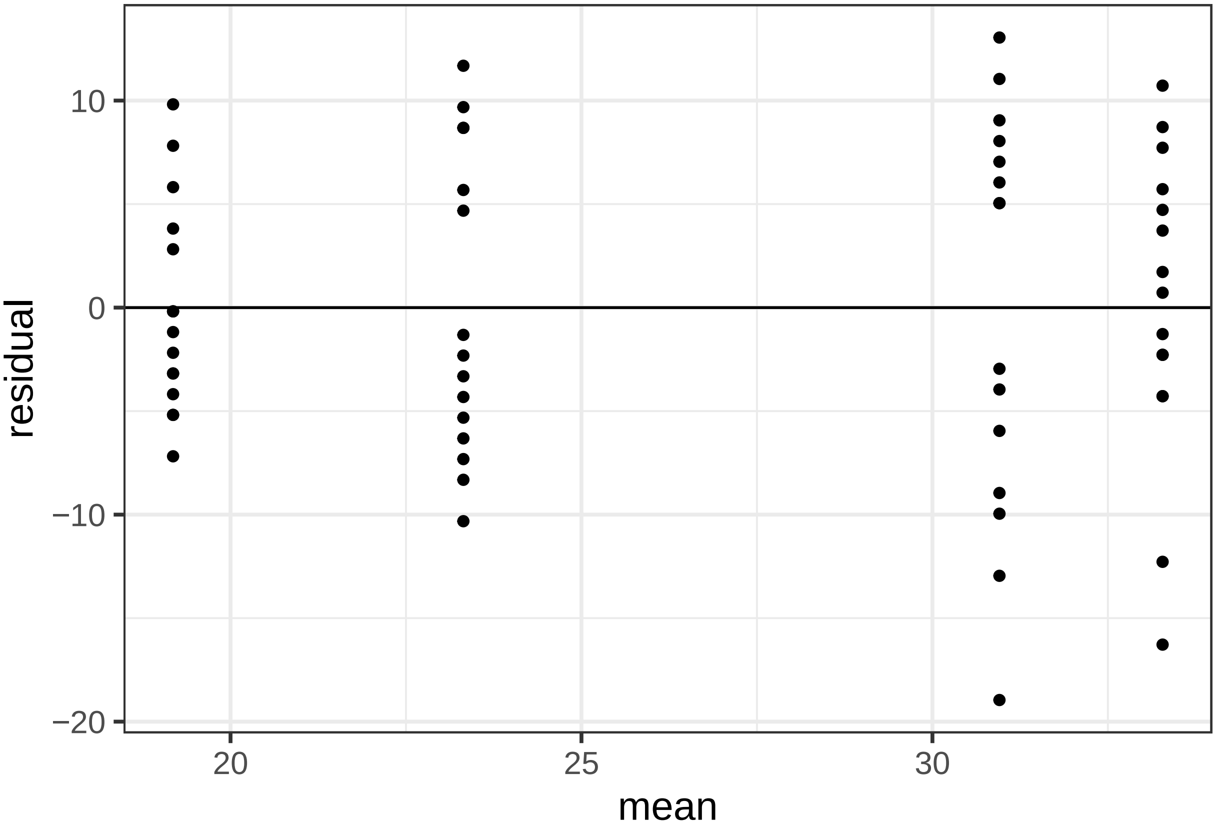
Table 8: WAIC Table

model	waic	deltaWAIC	weight
no_interaction	422.3176	2.183729	0.01252
interaction	432.8145	12.680610	0.00000
location_only	420.1339	0.000000	0.98748

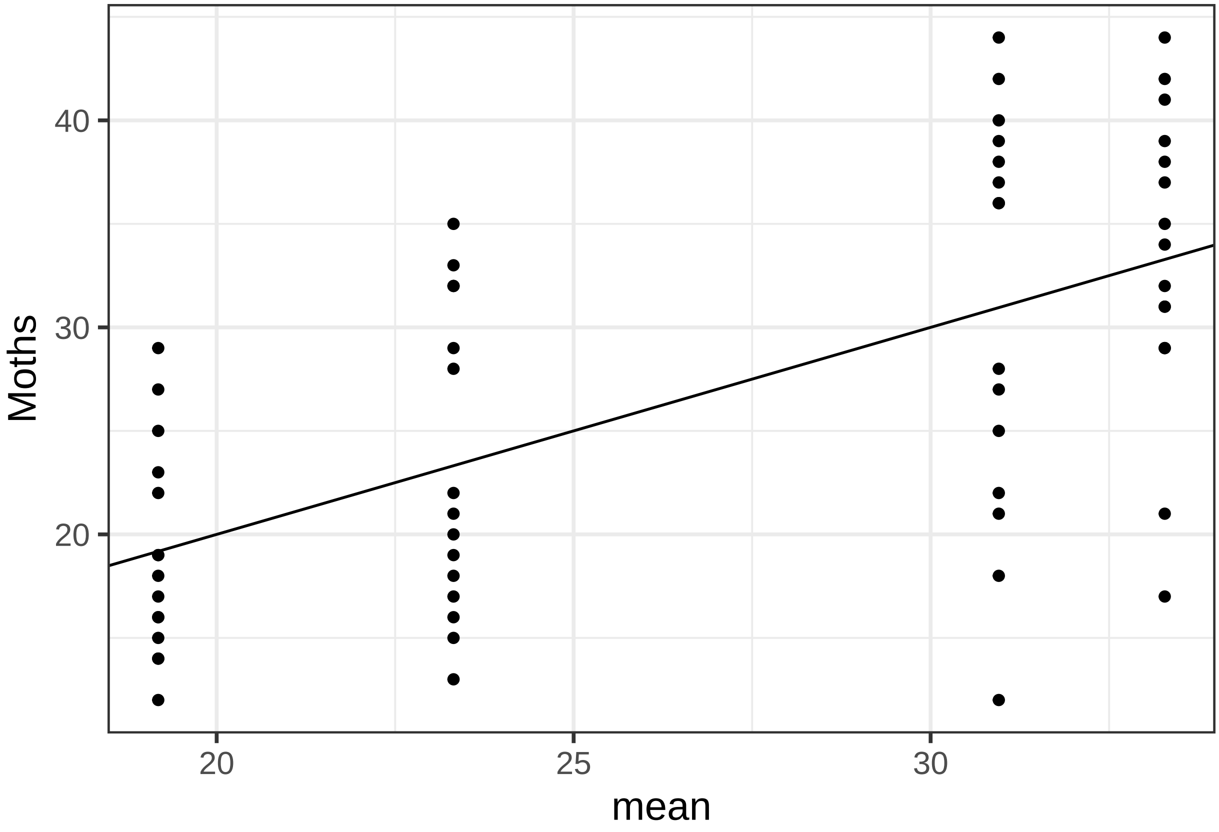
The model that only includes location is preferred.

```r
moth.loc.df <- data.frame(moth.loc.inla$summary.fitted.values) |>
  bind_cols(moth.data) |>
  mutate(residual = Moths-mean)


moth.loc.df |>
  ggplot(aes(x=mean,y=residual))+
  geom_point()+
  geom_abline(intercept=0,slope=0)
```

C) Plot the residuals vs. fitted values and qqnormal of the residuals from the WAIC best model for the moth data. Does the normal model seem to fit adequately?



```r
moth.loc.df |>
  ggplot(aes(x=mean,y=Moths))+
  geom_point()+
  geom_abline(intercept=0,slope=1)
```



The model seems to fit adequately, the residuals seem to be well distributed around zero across the predicted values. And the qqplot appears adequate.

D) Repeat the football models from part 2a and c in INLA, using default priors, and print the results with summary and autoplot. What did you get for the WAIC values and was the deltaWAIC result the same? *models*

```
#random effect

football.me.inla <- inla(Weight~1+f(Team,model = "iid"), data = football.data,
                         control.compute = list(dic = TRUE,
                                                waic = TRUE))

#no random effect

football.inla <- inla(Weight~1, data = football.data,
                         control.compute = list(dic = TRUE,
                                                waic = TRUE))
```

model summaries

```
#random effect

summary(football.me.inla)
```

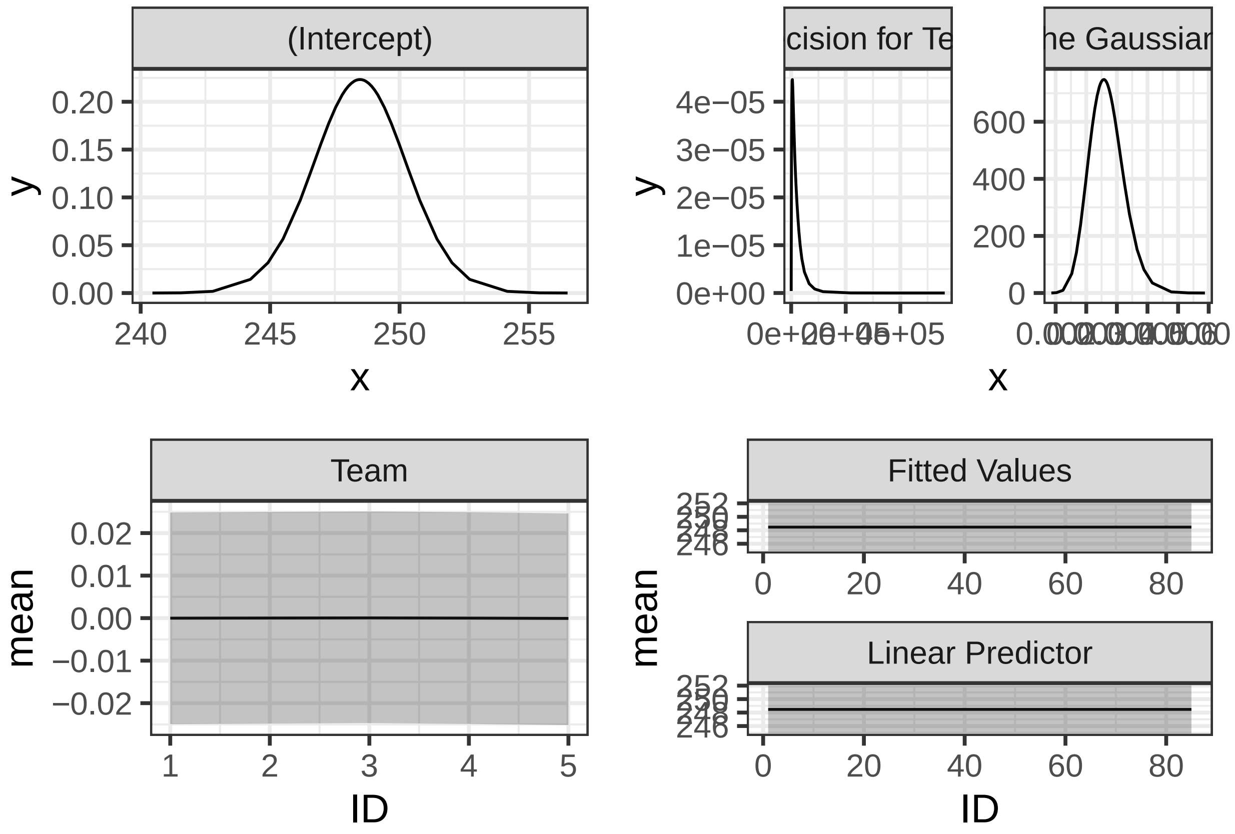
```
##
## Call:
##    c("inla.core(formula = formula, family = family, contrasts = contrasts, ", " data = data, quantile
##    quantiles, E = E, offset = offset, ", " scale = scale, weights = weights, Ntrials = Ntrials, stra
##    strata, ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose = verbose, ", " linc
##    lincomb, selection = selection, control.compute = control.compute, ", " control.predictor =
##    control.predictor, control.family = control.family, ", " control.inla = control.inla, control.fix
##    control.fixed, ", " control.mode = control.mode, control.expert = control.expert, ", " control.ha
##    control.hazard, control.lincomb = control.lincomb, ", " control.update = control.update, control.l
##    control.lp.scale, ", " control.pardiso = control.pardiso, only.hyperparam = only.hyperparam, ", "
##    = inla.call, inla.arg = inla.arg, num.threads = num.threads, ", " keep = keep, working.directory =
##    working.directory, silent = silent, ", " inla.mode = inla.mode, safe = FALSE, debug = debug, .pare
##    = .parent.frame)" )
## Time used:
##     Pre = 0.184, Running = 0.792, Post = 0.422, Total = 1.4
## Fixed effects:
##                mean    sd 0.025quant 0.5quant 0.975quant    mode kld
## (Intercept) 248.471 1.808    244.919  248.471    252.023 248.471   0
##
## Random effects:
##   Name     Model
##     Team IID model
##
## Model hyperparameters:
##
## Deviance Information Criterion (DIC) ...............: 722.74
## Deviance Information Criterion (DIC, saturated) ....: 89.14
## Effective number of parameters .....................: 1.86
##
## Watanabe-Akaike information criterion (WAIC) ...: 722.60
## Effective number of parameters .................: 1.68
##
## Marginal log-Likelihood:  -374.56
##  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)')
```

```r
#no random effect

summary(football.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, stra
##    strata, ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose = verbose, ", " linc
##    lincomb, selection = selection, control.compute = control.compute, ", " control.predictor =
##    control.predictor, control.family = control.family, ", " control.inla = control.inla, control.fix
##    control.fixed, ", " control.mode = control.mode, control.expert = control.expert, ", " control.ha
##    control.hazard, control.lincomb = control.lincomb, ", " control.update = control.update, control.l
##    control.lp.scale, ", " control.pardiso = control.pardiso, only.hyperparam = only.hyperparam, ", "
##    = inla.call, inla.arg = inla.arg, num.threads = num.threads, ", " keep = keep, working.directory =
##    working.directory, silent = silent, ", " inla.mode = inla.mode, safe = FALSE, debug = debug, .pare
##    = .parent.frame)" )
## Time used:
##     Pre = 0.165, Running = 0.118, Post = 0.346, Total = 0.629
## Fixed effects:
##                mean    sd 0.025quant 0.5quant 0.975quant    mode kld
## (Intercept) 248.471 1.809    244.915  248.471    252.026 248.471   0
##
## Model hyperparameters:
##
## Deviance Information Criterion (DIC) ...............: 722.99
## Deviance Information Criterion (DIC, saturated) ....: 89.39
## Effective number of parameters .....................: 1.99
##
## Watanabe-Akaike information criterion (WAIC) ...: 722.83
## Effective number of parameters .................: 1.78
##
## Marginal log-Likelihood:  -374.48
##  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)')
```

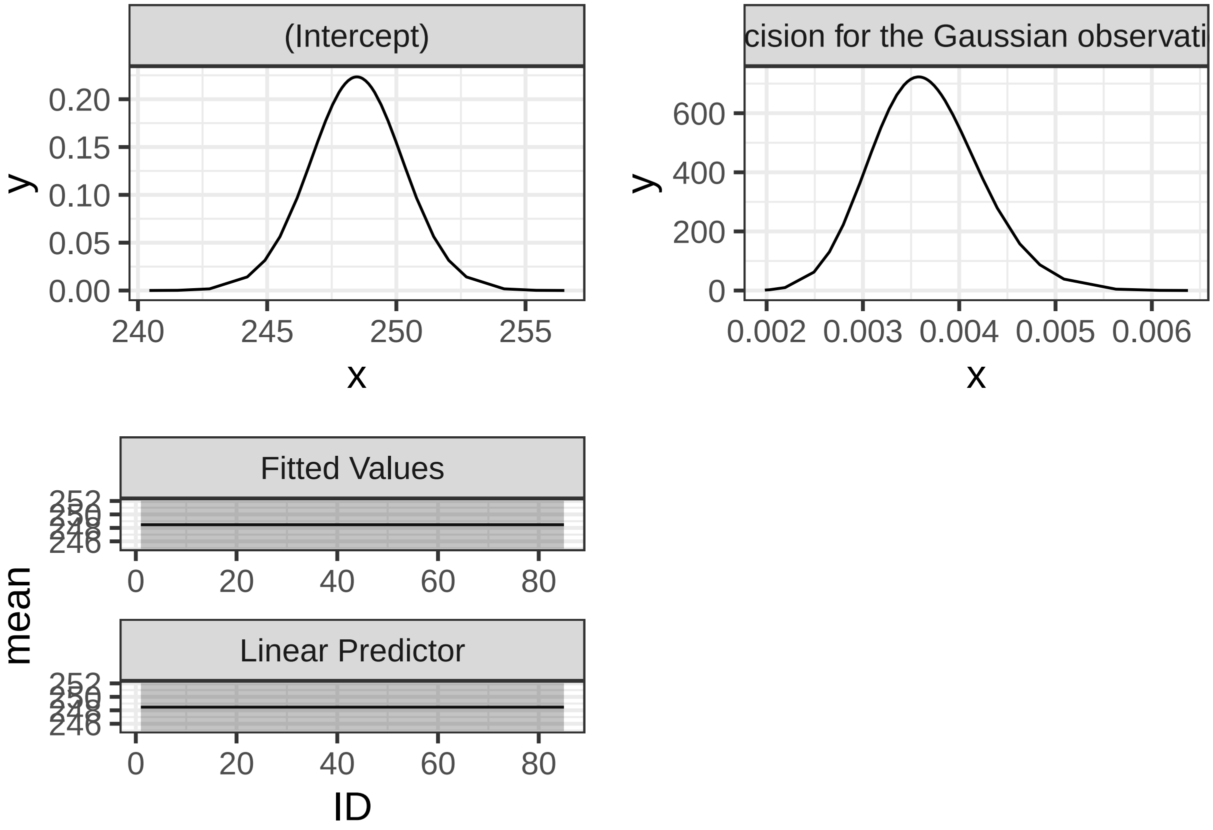
```
 *autoplots*
```

```r
#random effect

autoplot(football.me.inla)
```



```r
#no random effect

autoplot(football.inla)
```

```
## Warning in autoplot.inla(football.inla): Plot 3 selected in which, but no random effects to plot marg

## Warning in autoplot.inla(football.inla): Plot 3 will not be plotted.
```



```r
    ###WAIC

WAIC.table <- tibble(model = c("Random Effects", "No Random Effects"),
                    waic = c(football.me.inla$waic$waic,
                             football.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 9: WAIC Table

model	waic	deltaWAIC	weight
Random Effects	722.5970	0.0000000	0.61311
No Random Effects	722.8272	0.2302118	0.38689

model	waic	deltaWAIC	weight

The results were similar in that the both models were within 1 WAIC unit from each other. However, the INLA model said that the random effect model had the Lowest WAIC. Though this was only 0.230 WAIC units away from the model without random effects.