RE models

Barbara E. Mottey

3/3/2021

data

```
dat <- readRDS('C:/Users/sympl/Documents/UMass/msthesis/Data/completedata.rds')

to_factors <- c("fuel_bin", "gender", "residence", "wealth", "education", "marital_s", "region")
dat %<>% mutate_at(to_factors, funs(factor(.)))
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
sub2 <- dat[, 2:9]
#sub2$
intercept <- rep(1, nrow(sub2))

sub2<-cbind(intercept, sub2)
X.matrix2 <- sub2
```

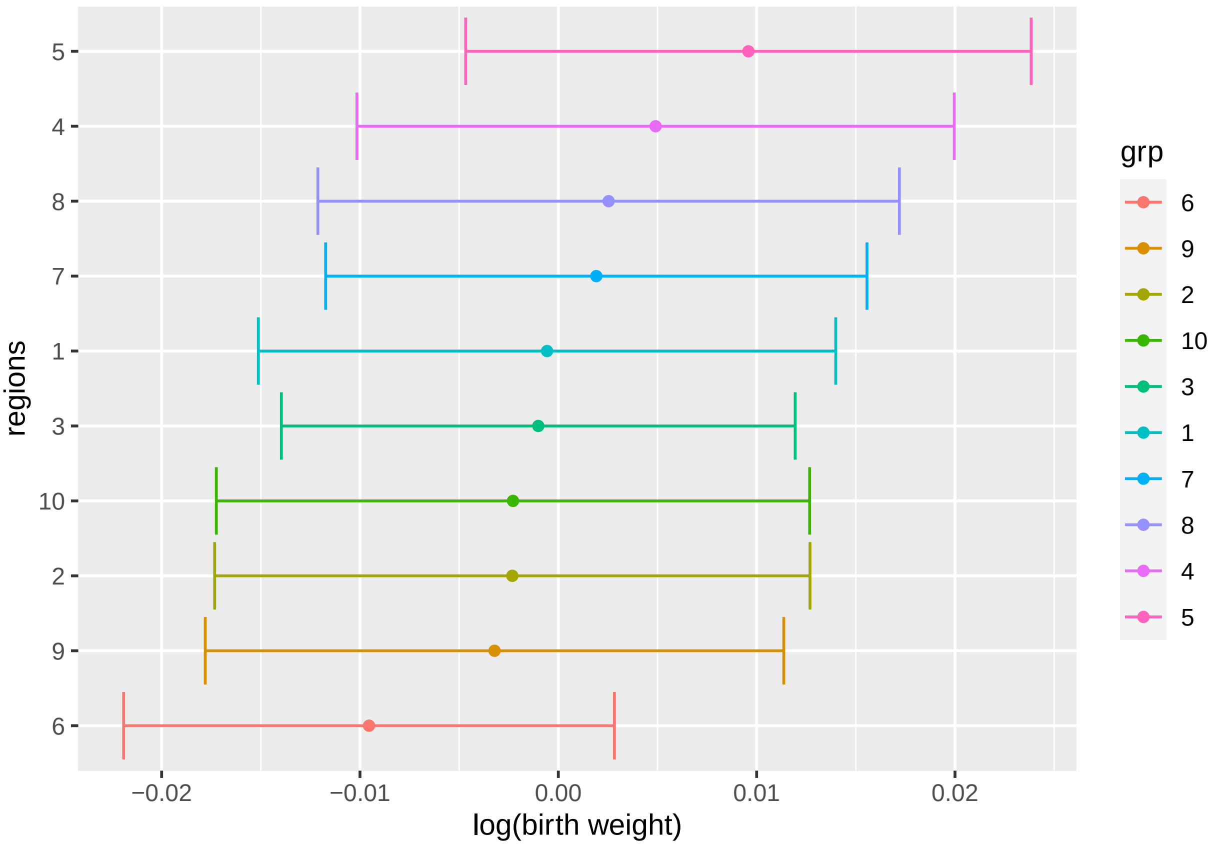
#LMER

```
mod1 <- lmer(log(c_weight)~as.factor(fuel_bin)+w_age+bmi+as.factor(gender)+as.factor(residence)+ as.fac


c(summary(mod1)$coefficient[2,1:2], confint(mod1, method="Wald")[4,1:2],nobs(mod1),"birthweight",
    paste0(round(summary(mod1)$coefficient[2,1:2][1],3)," (",
    round(confint(mod1, method="Wald")[4,1:2][1],3),", ",
    round(confint(mod1, method="Wald")[4,1:2][2],3),")"))
```

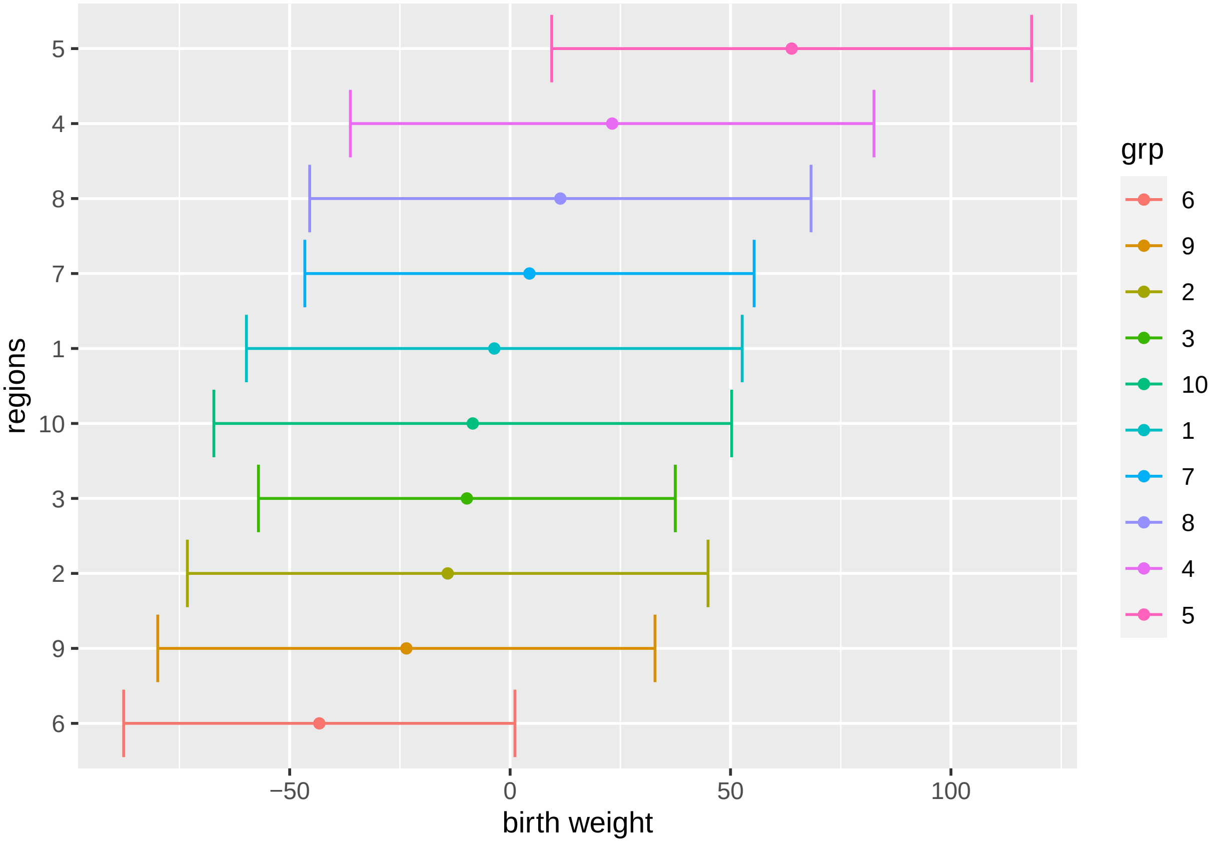
```
##                  Estimate               Std. Error                  2.5 %
##   "0.00847148681726682"     "0.012910463870867"   "-0.0168325573933382"
##                   97.5 %
##    "0.0337755310278718"                  "2960"           "birthweight"
##
## "0.008 (-0.017, 0.034)"
```

```
mod2 <- lmer(c_weight~as.factor(fuel_bin)+w_age+bmi+as.factor(gender)+as.factor(residence)+ as.factor(w
```

```
as.data.frame(ranef(mod1)) %>%
  ggplot(aes(y=grp, x= condval, col= grp))+
  geom_point()+
  geom_errorbarh(aes(xmin=condval+qnorm(0.025)*condsd, xmax= condval+qnorm(0.975)*condsd))+
  labs(main= "region intercepts and Confidence Intervals", x= "log(birth weight)", y= "regions")
```



```
as.data.frame(ranef(mod2)) %>%
  ggplot(aes(y=grp, x= condval, col= grp))+
  geom_point()+
  geom_errorbarh(aes(xmin=condval+qnorm(0.025)*condsd, xmax= condval+qnorm(0.975)*condsd))+
  labs(main= "region intercepts and Confidence Intervals", x= "birth weight", y= "regions")
```



stan

```
# data for stan
stanData <- list(
  # number of observations
  Nobs = nrow(dat),
  # number of predictors
  Npreds = dim(X.matrix2)[2],
  # number of regions
  J = nlevels(dat$region),
  # region indicators -- should be integers
  region = as.integer(dat$region),
  # response vector
  weight = dat$c_weight,
  # design matrix
  X = X.matrix2
  )
```

```
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())

model<- stan_model("C:/Users/sympl/Documents/UMass/msthesis/Model/stanbw.stan")
```

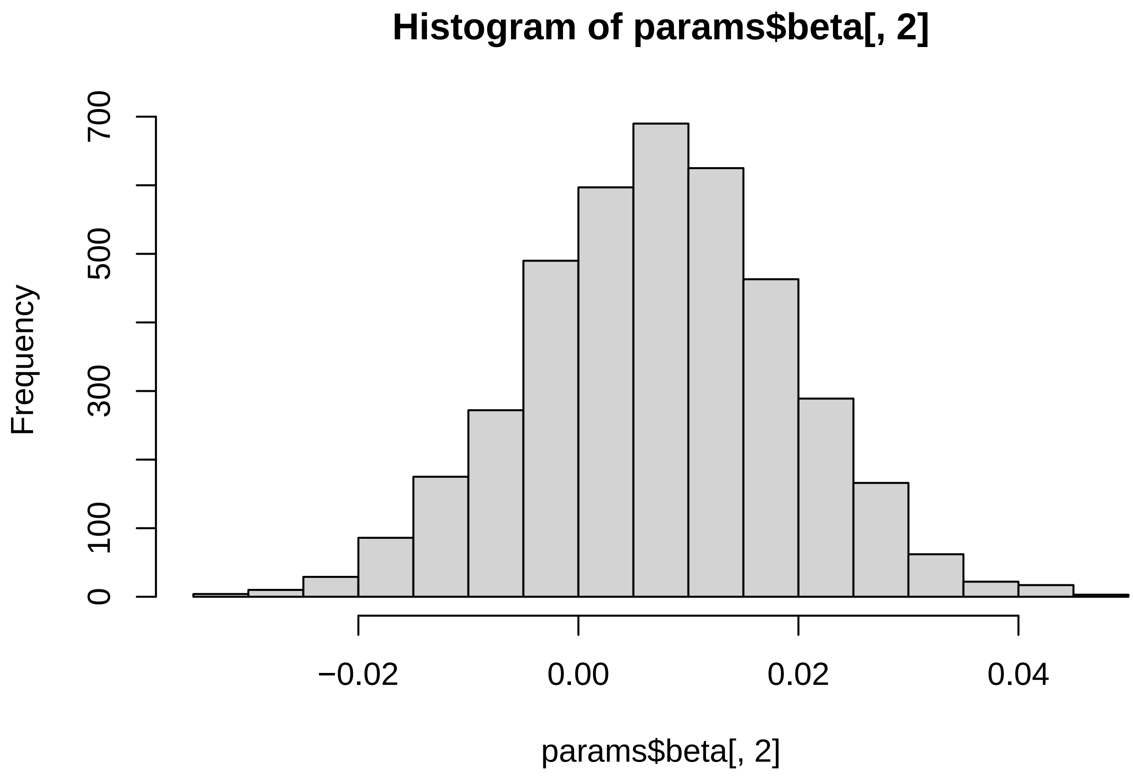
```r
fit1<- sampling(model, stanData,  iter=2000, chains=4)
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```r
params<- extract(fit1) #extract the parameters for graphing
```

```r
## histogram of average birthweight from cooking fuel
hist(params$beta[,2])
```



```r
labs=(main="posterior estimates from fuel use")


Betas <- (extract(fit1))$beta
apply(Betas, 2, mean)
```

```
## [1]  7.9538739340  0.0073818882  0.0018052447  0.0005511439 -0.0425619375
## [6]  0.0232335754  0.0088309064 -0.0028316316  0.0115264214
```

```r
##mean and CI of posterior estimate of cooking fuel
mean(Betas[, 2])
```

```
## [1] 0.007381888
```

```r
quantile(Betas[, 2], c(0.025,0.975))
```

```
##        2.5%       97.5% 
## -0.01646715  0.03005994
```

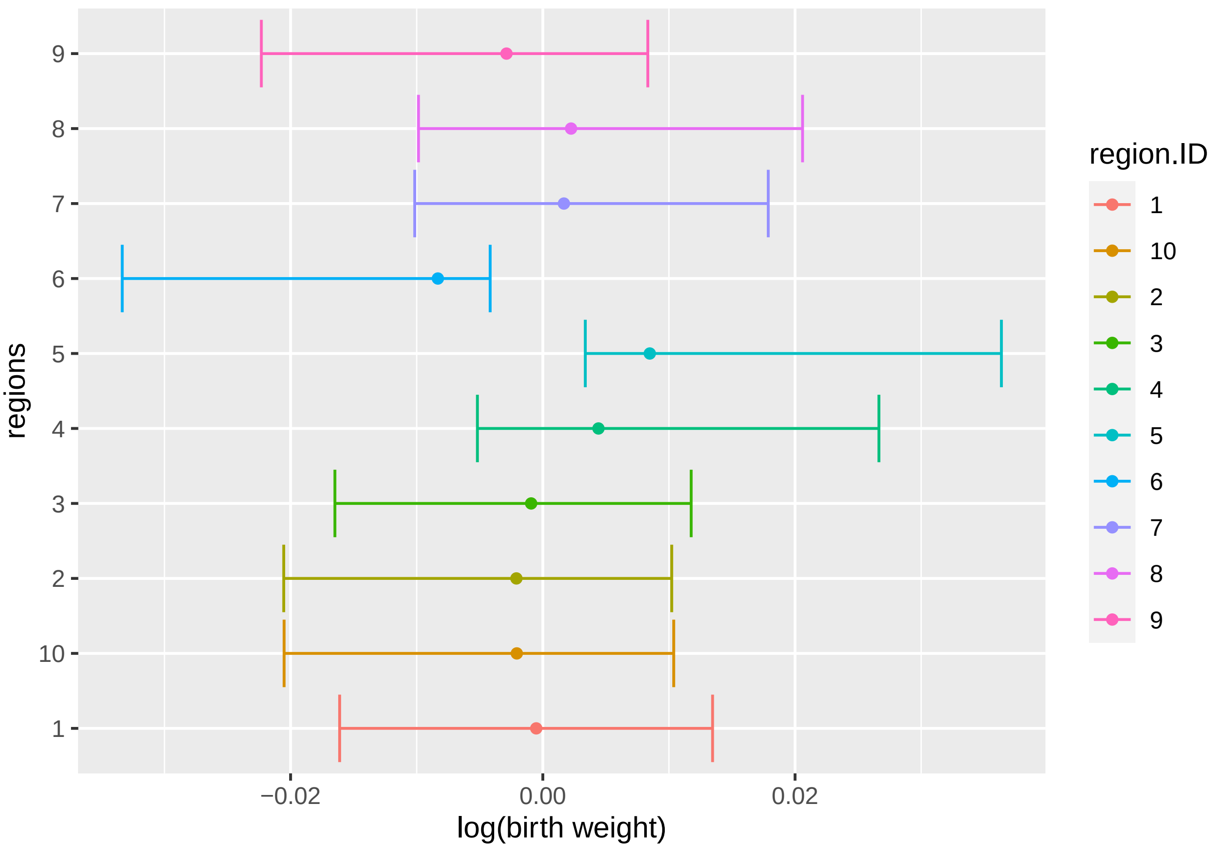
INLA

```r
library(INLA)
formula = log(c_weight)~1+fuel_bin+w_age+bmi+gender+residence+wealth+education+marital_s+ f(region, mode
result<-inla(formula, family = "gaussian", data=dat, control.predictor = list(compute = TRUE))

summary(result)
```

```
## 
## Call:
##    "inla(formula = formula, family = \"gaussian\", data = dat, 
##    control.predictor = list(compute = TRUE))" 
## Time used:
##     Pre = 2.43, Running = 5.2, Post = 1.71, Total = 9.34 
## Fixed effects:
##               mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept)  7.956 0.029      7.900    7.956      8.013  7.956   0
## fuel_bin1    0.008 0.013     -0.017    0.008      0.034  0.008   0
## w_age        0.002 0.001      0.001    0.002      0.003  0.002   0
## bmi          0.001 0.000      0.000    0.001      0.001  0.001   0
## gender2     -0.042 0.008     -0.057   -0.042     -0.027 -0.042   0
## residence2   0.022 0.010      0.002    0.022      0.041  0.022   0
## wealth2      0.027 0.014      0.000    0.027      0.054  0.027   0
## wealth3      0.042 0.014      0.014    0.042      0.070  0.042   0
## wealth4      0.025 0.015     -0.005    0.025      0.055  0.025   0
## wealth5      0.045 0.018      0.010    0.045      0.080  0.045   0
## education1  -0.020 0.012     -0.044   -0.020      0.004 -0.020   0
## education2  -0.011 0.011     -0.033   -0.011      0.010 -0.011   0
## education3  -0.010 0.022     -0.052   -0.010      0.033 -0.010   0
## marital_s1   0.013 0.012     -0.010    0.013      0.037  0.013   0
## 
## Random effects:
##   Name     Model
##     region IID model
## 
## Model hyperparameters:
##                                               mean       sd 0.025quant 0.5quant
## Precision for the Gaussian observations    23.39 6.13e-01      22.20    23.38
## Precision for region                    56821.66 1.70e+05    3587.24 20354.89
##                                         0.975quant     mode
## Precision for the Gaussian observations      24.61   23.37
## Precision for region                     332393.06 7169.08
##
## Expected number of effective parameters(stdev): 17.07(1.40)
## Number of equivalent replicates : 173.40
##
## Marginal log-Likelihood:  344.06
## Posterior marginals for the linear predictor and
##  the fitted values are computed
```

##plot of intercept

```
# df.ranef<-as.data.frame(result$summary.random)
#
# if(df.ranef[,1]==1){df.ranef$region.name="Western"}
# else if(df.ranef[,1]==2){df.ranef$region.name=="Central"}
# else if(df.ranef[,1]==3){region.name=="Greater Accra"}
# else if(df.ranef[,1]==4){region.name=="Volta"}
# else if(df.ranef[,1]==5){region.name=="Eastern"}
# else if(df.ranef[,1]==6){region.name=="Ashanti"}
# else if(df.ranef[,1]==7){region.name=="Brong Ahafo"}
# else if(df.ranef[,1]==8){region.name=="Northern"}
# else if(df.ranef[,1]==9){region.name=="Upper East"}
# else region.name=="Upper West"

as.data.frame(result$summary.random) %>%
  ggplot(aes(y=region.ID, x= region.mean, col= region.ID))+
  geom_point()+
  geom_errorbarh(aes(xmin=region.mean+region.0.025quant, xmax= region.mean+region.0.975quant))+
  labs(main= "region intercepts and Confidence Intervals", x= "log(birth weight)", y= "regions")
```



9
8
7
6
5
4
3
2
10
1
regions
−0.02
0.00
0.02
log(birth weight)
region.ID
1
10
2
3
4
5
6
7
8
9