# CAR Models

using Zimbabwe DHS data

*KM Wambui/E Musenge/Zvifadzo Matsena*

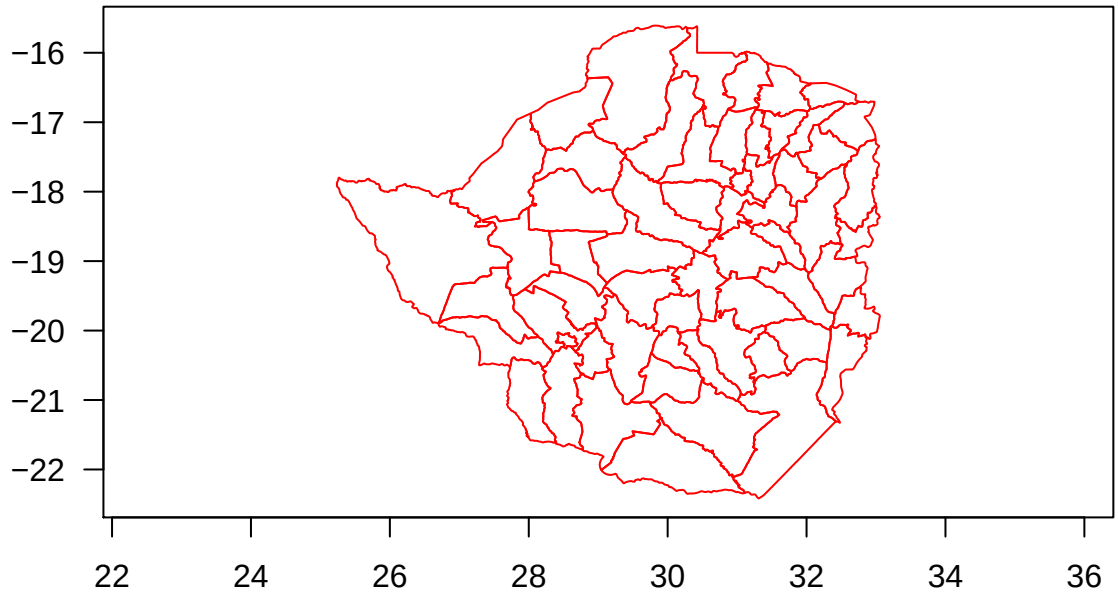
## Shape files in R

### Reading and exporting? the shape file

```
zim_shp <- maptools::readShapePoly("zim_shape/ZWE_adm2.shp", IDvar = "ID_2")
```

```
Warning: readShapePoly is deprecated; use rgdal::readOGR or sf::st_read
```

```
plot(zim_shp, border = "red", axes = TRUE, las = 1)
```

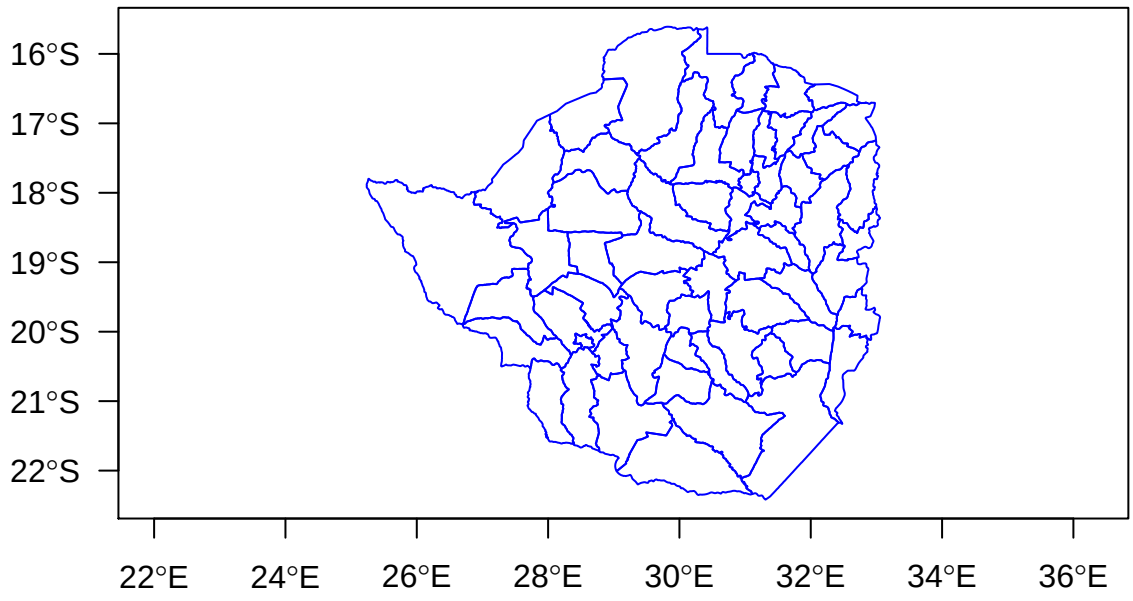



### What R is reccommending

```
zim_OGR <- rgdal::readOGR("zim_shape/ZWE_adm2.shp")
```

```
OGR data source with driver: ESRI Shapefile
Source: "H:\PhD_work\courses\WIts\CAR_model\zim_shape\ZWE_adm2.shp", layer: "ZWE_adm2"
with 60 features
It has 11 fields
Integer64 fields read as strings:  ID_0 ID_1 ID_2
```

```
plot(zim_OGR, border = "blue", axes = TRUE, las = 1)
```




We need to check our shape file to see if the spatial geometry is valid and the `gIsValid` function looks at each polygon, and makes sure it doesn't break any topological rules. Ideally, all of the geometry is valid and the total number of polygons that fail the validity tests is 0.

```
zim_valid <- data.table(valid = gIsValid(zim_OGR, byid = TRUE))
sum(zim_valid$valid == F)
```

```
[1] 0
```

## Export SPlus map format

Splus is the map file to use in Bayesian inference Using Gibbs Sampling (BUGS) softwares mostly OpenBugs or WinBUGS. `sp2WB` is the function exports an sp SpatialPolygons object into a S-Plus map format to be imported (**covered earlier**).

```
## saving the Ssplus map in zim_shp folder
maptools::sp2WB(map = as(zim_shp, "SpatialPolygons"), filename = "zim_shape/zim_spus")
```

## The adjacency matrix for the map

Remember the importance of CAR models , to adjust for the neighbouring effect. We extract the matrix to represent the structure of the neighbors. Can be done on OpenBUGS or in R (here we use the OpenBUGs extract)

```
## create the zim_nb map using poly2nb function
zim_nb <- spdep::poly2nb(zim_shp)
```

```
num <- sapply(zim_nb, length)
adj <- unlist(zim_nb)
sumNumNeigh <- length(unlist(zim_nb))

## neighbouring iformation from BUGS --> in the code file its included
## further down modelling
num = c(5, 7, 8, 4, 4, 8, 5, 3, 4, 7, 5, 5, 8, 6, 4, 8, 8, 8, 7, 4, 7, 5, 8,
    7, 6, 6, 5, 10, 3, 6, 7, 7, 7, 6, 6, 8, 7, 4, 5, 8, 4, 7, 6, 4, 8, 3, 6,
    6, 9, 2, 6, 6, 7, 6, 6, 6, 6, 5, 6, 5)
adj = c(52, 51, 47, 45, 40, 31, 26, 23, 19, 18, 13, 10, 35, 32, 25, 17, 7, 6,
    5, 4, 32, 7, 5, 3, 33, 32, 4, 3, 25, 22, 21, 19, 9, 8, 7, 3, 25, 8, 6, 4,
    3, 9, 7, 6, 22, 20, 8, 6, 21, 18, 16, 14, 13, 11, 2, 16, 14, 13, 12, 10,
    31, 30, 27, 13, 11, 31, 18, 16, 14, 12, 11, 10, 2, 24, 16, 15, 13, 11, 10,
    24, 20, 16, 14, 24, 21, 18, 15, 14, 13, 11, 10, 53, 35, 28, 26, 25, 23,
    19, 3, 24, 23, 21, 19, 16, 13, 10, 2, 25, 23, 21, 18, 17, 6, 2, 24, 22,
    15, 9, 24, 22, 19, 18, 16, 10, 6, 24, 21, 20, 9, 6, 31, 28, 26, 25, 19,
    18, 17, 2, 22, 21, 20, 18, 16, 15, 14, 23, 19, 17, 7, 6, 3, 31, 30, 28,
    23, 17, 2, 54, 30, 29, 28, 12, 57, 55, 54, 53, 31, 30, 27, 26, 23, 17, 54,
    39, 27, 54, 31, 28, 27, 26, 12, 30, 28, 26, 23, 13, 12, 2, 38, 36, 35, 33,
    5, 4, 3, 46, 38, 37, 36, 34, 32, 5, 60, 59, 58, 37, 36, 33, 53, 38, 36,
    32, 17, 3, 59, 53, 38, 37, 35, 34, 33, 32, 60, 58, 48, 46, 36, 34, 33, 36,
    35, 33, 32, 55, 54, 42, 41, 29, 57, 56, 52, 49, 45, 43, 42, 1, 47, 44, 42,
    39, 55, 45, 44, 43, 41, 40, 39, 57, 56, 55, 49, 42, 40, 47, 45, 42, 41,
    52, 51, 49, 47, 44, 42, 40, 1, 48, 37, 33, 51, 50, 45, 44, 41, 1, 58, 52,
    51, 49, 46, 37, 60, 59, 58, 56, 52, 48, 45, 43, 40, 51, 47, 52, 50, 48,
    47, 45, 1, 51, 49, 48, 45, 40, 1, 59, 57, 56, 36, 35, 28, 17, 55, 39, 30,
    29, 28, 27, 57, 54, 43, 42, 39, 28, 59, 57, 53, 49, 43, 40, 56, 55, 53,
    43, 40, 28, 60, 49, 48, 37, 34, 60, 56, 53, 49, 36, 34, 59, 58, 49, 37,
    34)
sumNumNeigh = 360
```

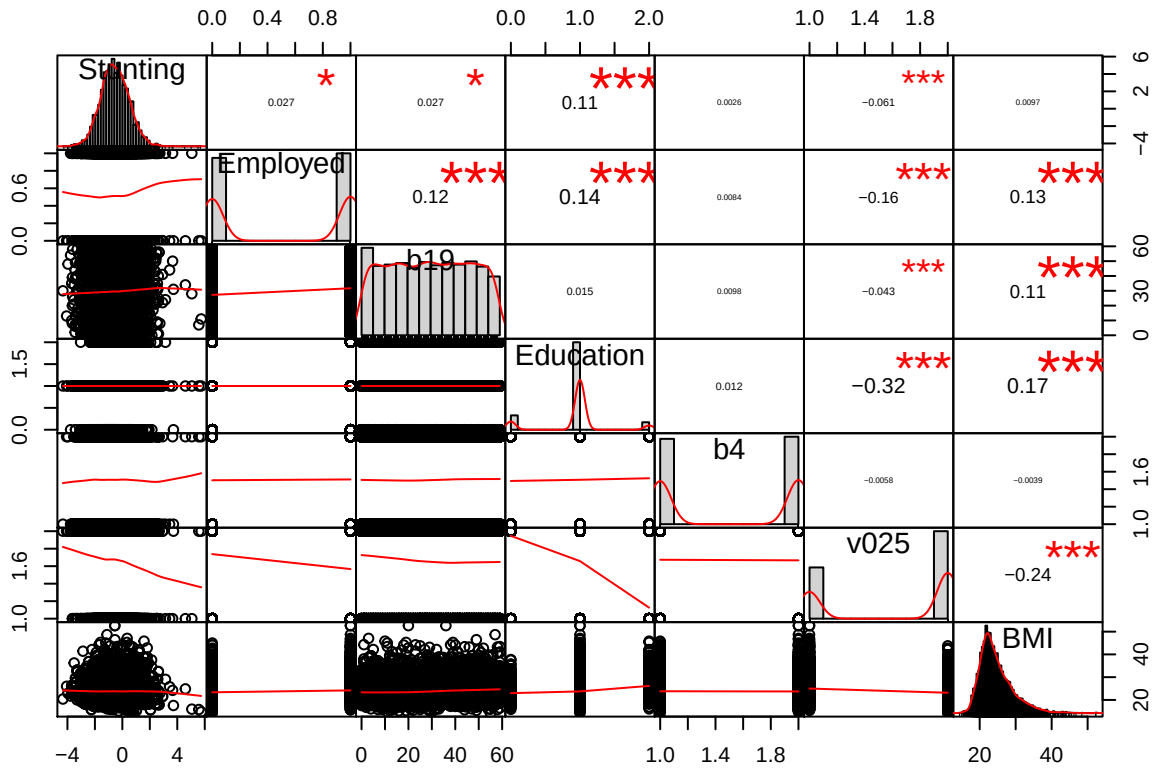
# Zimbabwe HAZ (Stunting) CAR Model

## Load the data

```
zim_child_data <- haven::read_dta("data/Data2.dta")
# dplyr::glimpse(zim_child_data , width = 5)
```

## Explolatory Data Analysis

```
## EDA
eda_data <- zim_child_data %>% select(Stunting, Employed, b19, Education, b4,
    v025, BMI)
PerformanceAnalytics::chart.Correlation(eda_data, histogram = TRUE, pch = 19)
```




## Linear regression model

```
## Define the formula of the model
form_fit <- Stunting ~ Employed + b19 + as.factor(Education) + b4 + v025 + BMI

## fit the model
lin_mod <- glm(form_fit, data = zim_child_data, family = gaussian(link = "identity"))
```

Extract the summary statistics from the model. What do the estimates mean?

```
summary(lin_mod)$coefficient
```

```
                          Estimate   Std. Error    t value     Pr(>|t|)
(Intercept)           -0.646520882 0.1142508269 -5.6587851 1.595528e-08
Employed               0.013977182 0.0275116631  0.5080457 6.114401e-01
b19                    0.001503499 0.0007767325  1.9356713 5.295526e-02
as.factor(Education)1  0.258544943 0.0406933905  6.3534874 2.260783e-10
as.factor(Education)2  0.487947342 0.0673593288  7.2439460 4.910341e-13
b4                     0.001661043 0.0266641938  0.0622949 9.503301e-01
v025                  -0.063267906 0.0299378549 -2.1133079 3.461591e-02
BMI                   -0.004092509 0.0030319814 -1.3497803 1.771378e-01
```

```
AIC(lin_mod)  # AIC => 17272
```

```
[1] 17272.36
```

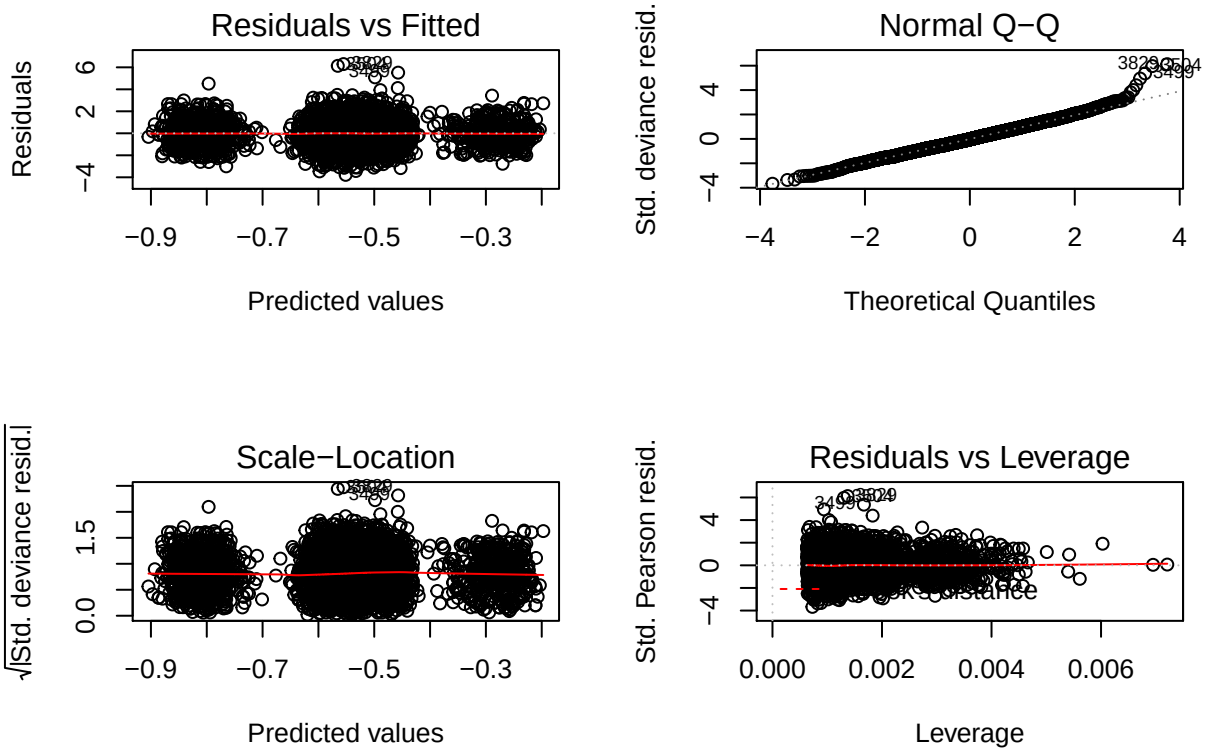
```
BIC(lin_mod)  # BIC => 17333
```

```
[1] 17332.6
```

## Check for the assumptions

We focus on the Residuals vs Fitted values and the Normal Q-Q. This will inform our decision on whether the outcome has met the assumptions to be used in further spatial models

```
par(mfrow = c(2, 2))  # display a unique layout for all graphs
plot(lin_mod)
```




```
par(mfrow = c(1, 1))
```

# CAR model in BUGS

## Select the variables of interest

What approaches can you use to select the variables of interest. - Stepwise regression 1^ - Regulization methods 2^

Here we have already selected the variables to use and we use complete case data.

```
zim_child_model <- zim_child_data %>% select(id_2, name_1, name_2, Stunting,
    Employed, b19, Education, b4, v025, BMI)
zim_child_model <- zim_child_model[complete.cases(zim_child_model), ]
glimpse(zim_child_model)
```

```
Observations: 5,962
Variables: 10
$ id_2      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1...
$ name_1    <chr> "Bulawayo", "Harare", "Manicaland", "Manicaland", "M...
$ name_2    <chr> "Bulawayo", "Harare", "Buhera", "Chimanimani", "Chip...
$ Stunting  <dbl> -1.93, -0.86, -1.34, 0.80, 1.86, -0.32, 1.01, -1.06,...
$ Employed  <dbl> 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0...
$ b19       <dbl> 13, 44, 18, 27, 1, 28, 5, 16, 8, 32, 59, 36, 23, 48,...
$ Education <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1...
$ b4        <dbl+lbl> 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 2, 1, 1, ...
$ v025      <dbl+lbl> 2, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, ...
$ BMI       <dbl> 21.15, 37.29, 28.26, 21.70, 19.64, 19.56, 30.28, 28....
```

Check whether the ID_2 on the shape file corresponds to what you have in your data

```
table(zim_shp@data$ID_2 %in% zim_child_model$ID_2)
```

```
Warning: Unknown or uninitialised column: 'ID_2'.
```

```
FALSE
   60
```

```
table(zim_shp@data$NAME_2 %in% zim_child_model$name_1)
```

```
FALSE  TRUE
   57     3
```

## Creating a data set to use in BUGS

We create a list in R with all the variables of interest.

```
data_model <- list(N = length(zim_child_model$Stunting), adj = adj, num = num,
    sumNumNeigh = sumNumNeigh, Stunting = zim_child_model$Stunting, Employed = as.numeric(zim_child_mod
    Education = zim_child_model$Education, b4 = as.numeric(zim_child_model$b4),
    v025 = as.numeric(zim_child_model$v025), BMI = as.numeric(zim_child_model$BMI),
    b19 = as.numeric(zim_child_model$b19), id_2 = zim_child_model$id_2)
names(data_model)
```

```
 [1] "N"           "adj"         "num"         "sumNumNeigh" "Stunting"
 [6] "Employed"    "Education"   "b4"          "v025"        "BMI"
[11] "b19"         "id_2"
```

## Our BUGS code

The code is saved as `bugs_models.txt` in our working directory and contains

```
model {
#Likelihood
for( i in 1 : N ) {
            Stunting[i] ~ dnorm(mu[i], tau)
            mu[i]<-beta[1]+
              beta[2]*equals(Education[i],1)+
              beta[3]*equals(Education[i],2)+
              beta[4]*equals(b4[i],2)+
              beta[5]*equals(v025[i],2)+
              beta[6]*equals(Employed[i],1)+
            beta[7]*BMI[i]+beta[8]*b19[i]+
               u[id_2[i]] + Phi[id_2[i]]
}
#Priors
for (i in 1:8)
{
 beta[i]~dnorm(0.0, 0.001)
}
###### Unstructured groups
for (k in 1:60)
 {
 u[k] ~ dnorm(0,tauu)
}
###### CAR prior distribution for structured random effects:
# Bivariate CAR Prior for Phi -- Spatial Main Effects
Phi[1:60] ~ car.normal(adj[], weights[], num[], tau)     # num specifies no. of neighbors
for(i in 1:sumNumNeigh){weights[i] <- 1}
###### Hyperpriors
tauu ~ dgamma(0.01, 0.01)
tau  ~ dgamma(0.5, 0.0005)
#tauVal <- loggam(tau)
tauh  ~ dgamma(0.5, 0.0005)
}
```

## Defining the important parameters

```
### defining the initials Initial Inits
inits_Vals <- function() {
    list(beta = c(0, 0, 0, 0, 0, 0, 0, 0))
}

## parameters we want to monitor in R
params <- c("beta", "Phi")
```

## Running the CAR model from R

Here we use the `R2OpenBUGS` package using the `bugs` function to push the model to OpenBUGS.

- `data` is the data we defined to use
- `model.file="bugs_models.txt"` is the file in our working directory with the BUGS code `-inits_Vals1` = are the initial points where the different parameters in our model start for the MCMC. Mostly we

can use the mean of the variable but here we use 0 for only the beta parameters. (Figure out how to initialize for the other parameters)

- `parameters.to.save` - parameters we want to save from the model. here we save the regression `beta` coefficients and the location (`id_2`) mean estimate of HAZ
- `n.chains` - specifying the number of markov chain. (Figure out how to fit for 2 chains)
- `n.iter=10000` - specifying the number of iterations for our MCMC. We have specified 10000 (reduce to 1000 for this exercise)
- `n.burnin=2000` - the values to burn in , can be guided by the diagnostic plots. We have burned in 2000 (reduce to 200 for this exercise)
- `n.thin=10` - the values to thin to reduce autocorrelation

```
model_bugs <-  R2OpenBUGS::bugs(data = data_model,
                                model.file="bugs_models.txt",
                                inits=inits_Vals,
                                parameters.to.save = params,
                                n.chains=1,n.iter=10000,
                                n.burnin=2000,n.thin=10,codaPkg=T,digits = 5,
                                debug = F,DIC=TRUE,clearWD = TRUE,
                               ## uncomment below for WinBUGs
                               ##bugs.dir = "C:/Program Files (x86)/WinBUGS14/",
                                working.directory = file.path(paste0(getwd(),"codas")))
```

Since I had run the model above , I read in the saved coda or ourput

```
coda_res <- R2OpenBUGS::read.bugs("codas/CODAchain1.txt")
```

## Get the posterior means of the model parameters

```
##
summary_model <- summary(coda_res)
summary_model$statistics
```

```
                 Mean           SD     Naive SE Time-series SE
Phi[1]   -3.765914e-01 0.2542489707 2.842590e-03   6.988551e-03
Phi[2]   -1.518945e-01 0.2423305028 2.709337e-03   6.325255e-03
Phi[3]    1.538581e-01 0.3760701237 4.204592e-03   4.634338e-03
Phi[4]    5.212500e-01 0.5009083896 5.600326e-03   5.600326e-03
Phi[5]    6.928717e-01 0.4959668072 5.545077e-03   5.545077e-03
Phi[6]    6.591962e-02 0.3701431996 4.138327e-03   4.285841e-03
Phi[7]    3.395771e-01 0.4469118192 4.996626e-03   5.524545e-03
Phi[8]   -6.093684e-02 0.5366706040 6.000160e-03   6.124765e-03
Phi[9]   -3.620516e-02 0.2652499084 2.965584e-03   7.182650e-03
Phi[10]  -2.021244e-01 0.3963309321 4.431115e-03   4.930295e-03
Phi[11]  -1.975121e-01 0.4504031960 5.035661e-03   5.196573e-03
Phi[12]  -2.140306e-02 0.4427722471 4.950344e-03   5.323458e-03
Phi[13]   6.707042e-03 0.3663415860 4.095823e-03   4.232209e-03
Phi[14]  -1.829462e-01 0.4198690220 4.694278e-03   5.203173e-03
Phi[15]   2.119833e-01 0.4907741849 5.487022e-03   5.606790e-03
Phi[16]  -8.158392e-02 0.2444796836 2.733366e-03   6.700126e-03
Phi[17]   1.622997e-01 0.3629297359 4.057678e-03   4.148718e-03
Phi[18]  -9.850619e-02 0.3674503546 4.108220e-03   4.791730e-03
Phi[19]   1.846938e-01 0.3805038930 4.254163e-03   4.254163e-03
```

| | | | | |
|---|---|---|---|---|
| Phi[20] | 1.092420e-01 | 0.4902961096 | 5.481677e-03 | 5.599535e-03 |
| Phi[21] | -1.475761e-01 | 0.3896873172 | 4.356837e-03 | 4.459420e-03 |
| Phi[22] | 1.023683e-01 | 0.4504117726 | 5.035757e-03 | 5.344191e-03 |
| Phi[23] | 3.354255e-02 | 0.3548402199 | 3.967234e-03 | 4.073711e-03 |
| Phi[24] | 6.290283e-02 | 0.4041322855 | 4.518336e-03 | 4.678974e-03 |
| Phi[25] | 1.430993e-03 | 0.2500748991 | 2.795922e-03 | 6.260265e-03 |
| Phi[26] | 1.148181e-01 | 0.4053379064 | 4.531816e-03 | 4.857448e-03 |
| Phi[27] | -1.612901e-02 | 0.4607564325 | 5.151414e-03 | 5.244129e-03 |
| Phi[28] | 5.881298e-02 | 0.3362594851 | 3.759495e-03 | 3.932434e-03 |
| Phi[29] | -7.174692e-02 | 0.5641920843 | 6.307859e-03 | 6.307859e-03 |
| Phi[30] | -9.651412e-02 | 0.4159975798 | 4.650994e-03 | 4.747423e-03 |
| Phi[31] | -1.217376e-01 | 0.2451312459 | 2.740651e-03 | 5.823341e-03 |
| Phi[32] | 3.296977e-01 | 0.3988355808 | 4.459117e-03 | 4.610015e-03 |
| Phi[33] | 3.785181e-01 | 0.3918446112 | 4.380956e-03 | 4.493958e-03 |
| Phi[34] | 1.616594e-01 | 0.4141040531 | 4.629824e-03 | 4.839224e-03 |
| Phi[35] | 1.726983e-01 | 0.4075364147 | 4.556396e-03 | 4.738065e-03 |
| Phi[36] | 3.128653e-01 | 0.3699893192 | 4.136606e-03 | 4.488109e-03 |
| Phi[37] | 3.271229e-01 | 0.3984058706 | 4.454313e-03 | 4.720380e-03 |
| Phi[38] | -1.039133e-02 | 0.2672951255 | 2.988450e-03 | 8.086331e-03 |
| Phi[39] | 6.304502e-02 | 0.4624881852 | 5.170775e-03 | 5.170775e-03 |
| Phi[40] | -1.513209e-01 | 0.3597100116 | 4.021680e-03 | 4.708415e-03 |
| Phi[41] | -3.209877e-01 | 0.4970568598 | 5.557265e-03 | 5.557265e-03 |
| Phi[42] | -2.995958e-01 | 0.3941870813 | 4.407146e-03 | 4.862660e-03 |
| Phi[43] | -2.131190e-01 | 0.4224401270 | 4.723024e-03 | 5.299862e-03 |
| Phi[44] | -3.469810e-01 | 0.4910037844 | 5.489589e-03 | 5.489589e-03 |
| Phi[45] | -2.313117e-01 | 0.2436576607 | 2.724175e-03 | 6.299059e-03 |
| Phi[46] | 3.029285e-01 | 0.5505737829 | 6.155602e-03 | 6.155602e-03 |
| Phi[47] | -4.118183e-01 | 0.4238411543 | 4.738688e-03 | 5.132431e-03 |
| Phi[48] | -1.071276e-01 | 0.4189941277 | 4.684497e-03 | 4.821841e-03 |
| Phi[49] | 2.315206e-02 | 0.3476368018 | 3.886698e-03 | 4.130036e-03 |
| Phi[50] | 5.156208e-02 | 0.6404069662 | 7.159968e-03 | 7.159968e-03 |
| Phi[51] | -4.829538e-01 | 0.4175068548 | 4.667869e-03 | 5.302005e-03 |
| Phi[52] | -1.639462e-01 | 0.2636472959 | 2.947666e-03 | 7.416184e-03 |
| Phi[53] | 2.193514e-01 | 0.3873588984 | 4.330804e-03 | 4.367040e-03 |
| Phi[54] | 1.189655e-01 | 0.4357734315 | 4.872095e-03 | 4.872095e-03 |
| Phi[55] | 6.846071e-02 | 0.4212380637 | 4.709585e-03 | 4.709585e-03 |
| Phi[56] | -1.604377e-01 | 0.4145032188 | 4.634287e-03 | 4.634287e-03 |
| Phi[57] | -1.904300e-01 | 0.4179150663 | 4.672432e-03 | 4.672432e-03 |
| Phi[58] | -2.488218e-01 | 0.4497626129 | 5.028499e-03 | 5.028499e-03 |
| Phi[59] | -1.538202e-02 | 0.4128545608 | 4.615854e-03 | 4.774536e-03 |
| Phi[60] | -1.342823e-01 | 0.2565212093 | 2.867994e-03 | 6.256363e-03 |
| beta[1] | -5.512804e-01 | 0.1186765033 | 1.326844e-03 | 4.202157e-03 |
| beta[2] | 2.555349e-01 | 0.0412779059 | 4.615010e-04 | 6.292783e-04 |
| beta[3] | 4.807857e-01 | 0.0686230067 | 7.672285e-04 | 8.915790e-04 |
| beta[4] | 1.919287e-03 | 0.0264677085 | 2.959180e-04 | 3.083280e-04 |
| beta[5] | -5.516357e-02 | 0.0325889208 | 3.643552e-04 | 5.227015e-04 |
| beta[6] | 1.213160e-02 | 0.0275125707 | 3.075999e-04 | 3.075999e-04 |
| beta[7] | -3.201295e-03 | 0.0030474051 | 3.407102e-05 | 8.635404e-05 |
| beta[8] | 1.423969e-03 | 0.0007716414 | 8.627213e-06 | 8.767351e-06 |
| deviance | 1.719229e+04 | 8.7944666029 | 9.832513e-02 | 1.044767e-01 |

## Model diagnostics

Here we perform model diagnostics using the code below. It generates posterior estimates plots and other diagnostic plots

```
## diagnostics
dir.create("results/str_unstru/figures", recursive = T)
mcmcplots::mcmcplot(coda_res, dir = file.path(getwd(), "results/str_unstru/figures"),
    regex = "beta|Phi")
```

## Ploting our estimated to a map

We extract the spatial estimates from the model using `summary_model$statistics` and merge with the shape file and plot

```
## add the mean to a plot
df_model <- as.data.frame(summary_model$statistics) %>%
  tibble::rownames_to_column("id_var")


## select spatial parameters only
df_model_spat <- df_model %>%
  ##remain with Phi var
  filter(grepl("Phi",id_var)) %>%
  mutate(ID_2=gsub("Phi\\[","",id_var),
         ID_2=gsub("\\]","",ID_2))
glimpse(df_model_spat)
```

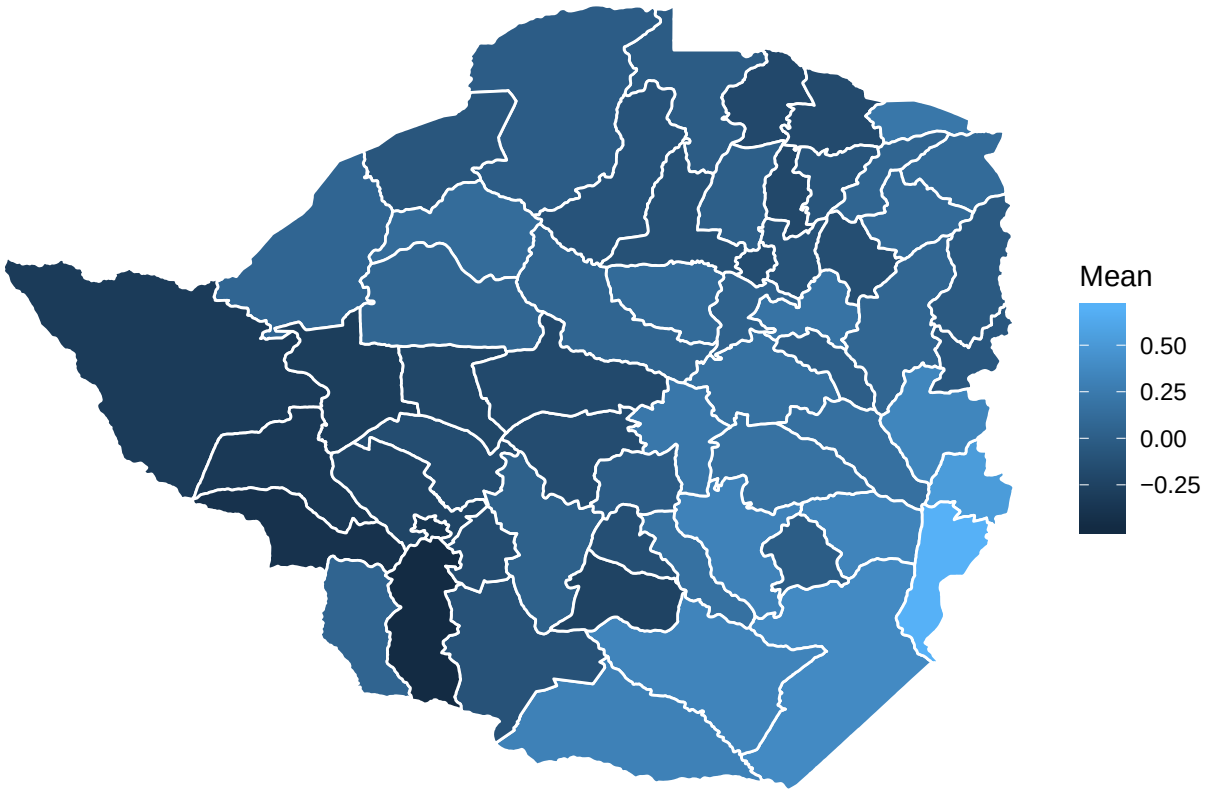
```
Observations: 60
Variables: 6
$ id_var           <chr> "Phi[1]", "Phi[2]", "Phi[3]", "Phi[4]", "Phi[...
$ Mean             <dbl> -0.376591393, -0.151894530, 0.153858141, 0.52...
$ SD               <dbl> 0.2542490, 0.2423305, 0.3760701, 0.5009084, 0...
$ `Naive SE`       <dbl> 0.002842590, 0.002709337, 0.004204592, 0.0056...
$ `Time-series SE` <dbl> 0.006988551, 0.006325255, 0.004634338, 0.0056...
$ ID_2             <chr> "1", "2", "3", "4", "5", "6", "7", "8", "9", ...
```

```
## plot the mean estimates on the map
zim_shp@data <- zim_shp@data
zim_shp_df <- broom::tidy(zim_shp, region = "ID_2")
zim_shp_df <- zim_shp_df %>%
  left_join(df_model_spat, by=c("id"="ID_2"))

glimpse(zim_shp_df)
```

```
Observations: 30,130
Variables: 12
$ long             <dbl> 28.61305, 28.60440, 28.57862, 28.55219, 28.54...
$ lat              <dbl> -20.23587, -20.22540, -20.22598, -20.22138, -...
$ order            <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...
$ hole             <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...
$ piece            <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
$ group            <chr> "1.1", "1.1", "1.1", "1.1", "1.1", "1.1", "1....
$ id               <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1", ...
$ id_var           <chr> "Phi[1]", "Phi[1]", "Phi[1]", "Phi[1]", "Phi[...
$ Mean             <dbl> -0.3765914, -0.3765914, -0.3765914, -0.376591...
```

```
$ SD                <dbl> 0.254249, 0.254249, 0.254249, 0.254249, 0.254...
$ `Naive SE`        <dbl> 0.00284259, 0.00284259, 0.00284259, 0.0028425...
$ `Time-series SE` <dbl> 0.006988551, 0.006988551, 0.006988551, 0.0069...
```

```
p1 <- ggplot() +
  geom_polygon(data = zim_shp_df, aes(x = long, y = lat,
                                      group = group,
                                      fill = Mean), colour = "white") + theme_void()
p1
```




# CAR model in INLA

It uses the Integrated Nested Laplace Approximation, a deterministic Bayesian method [1] and a very nice book here by Marta Blangiardo [2].

## Normal linear model in INLA

In the formula below 1 means that the model includes the intercept and

```
form_fit <-  Stunting ~1+ Employed +b19  +
  as.factor(Education) + b4+ v025+ BMI


model_linear <- inla(form_fit,
```

[1] http://www.r-inla.org/download
[2] http://www.r-inla.org/books

```
                     family="gaussian",
                     data=zim_child_model,
                    control.compute=list(dic=TRUE, waic=TRUE))
```

### Extracting the model summaries

Here we compare our glm model with the linear model from INLA, are they similar?

```
summary(model_linear)
```

```
Call:
c("inla(formula = form_fit, family = \"gaussian\", data = zim_child_model, ",  "    control.compute = li

Time used:
 Pre-processing    Running inla Post-processing           Total
         1.9818          6.0923          0.4099          8.4840

Fixed effects:
                         mean     sd 0.025quant 0.5quant 0.975quant
(Intercept)           -0.6465 0.1142    -0.8708  -0.6465    -0.4224
Employed               0.0140 0.0275    -0.0400   0.0140     0.0679
b19                    0.0015 0.0008     0.0000   0.0015     0.0030
as.factor(Education)1  0.2585 0.0407     0.1786   0.2585     0.3384
as.factor(Education)2  0.4879 0.0674     0.3557   0.4879     0.6201
b4                     0.0017 0.0267    -0.0507   0.0017     0.0540
v025                  -0.0633 0.0299    -0.1220  -0.0633    -0.0045
BMI                   -0.0041 0.0030    -0.0100  -0.0041     0.0019
                         mode kld
(Intercept)           -0.6465   0
Employed               0.0140   0
b19                    0.0015   0
as.factor(Education)1  0.2585   0
as.factor(Education)2  0.4879   0
b4                     0.0017   0
v025                  -0.0633   0
BMI                   -0.0041   0

The model has no random effects

Model hyperparameters:
                                          mean     sd 0.025quant 0.5quant
Precision for the Gaussian observations 0.9444 0.0171     0.9112   0.9443
                                        0.975quant   mode
Precision for the Gaussian observations     0.9783 0.9441

Expected number of effective parameters(std dev): 8.034(6e-04)
Number of equivalent replicates : 742.05

Deviance Information Criterion (DIC) ...............: 17272.44
Deviance Information Criterion (DIC, saturated) ....: 5973.89
Effective number of parameters .....................: 9.072

Watanabe-Akaike information criterion (WAIC) ...: 17272.83
```

```
Effective number of parameters .................: 9.444

Marginal log-Likelihood:  -8697.69
Posterior marginals for linear predictor and fitted values computed
```

```
summary(lin_mod)
```

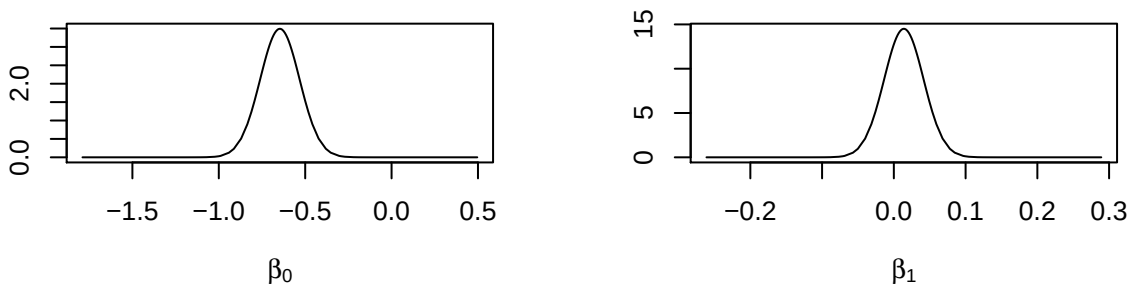
```
Call:
glm(formula = form_fit, family = gaussian(link = "identity"),
    data = zim_child_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.7788  -0.6882  -0.0303   0.6685   6.2951

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           -0.6465209  0.1142508  -5.659 1.60e-08 ***
Employed               0.0139772  0.0275117   0.508   0.6114
b19                    0.0015035  0.0007767   1.936   0.0530 .
as.factor(Education)1  0.2585449  0.0406934   6.353 2.26e-10 ***
as.factor(Education)2  0.4879473  0.0673593   7.244 4.91e-13 ***
b4                     0.0016610  0.0266642   0.062   0.9503
v025                  -0.0632679  0.0299379  -2.113   0.0346 *
BMI                   -0.0040925  0.0030320  -1.350   0.1771
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 1.059208)

    Null deviance: 6399.5  on 5961  degrees of freedom
Residual deviance: 6306.5  on 5954  degrees of freedom
AIC: 17272

Number of Fisher Scoring iterations: 2
```

Extracting the summary of the fixed effects

```
round(model_linear$summary.fixed[, 1:5], 3)
```

```
                       mean    sd 0.025quant 0.5quant 0.975quant
(Intercept)          -0.647 0.114     -0.871   -0.647     -0.422
Employed              0.014 0.028     -0.040    0.014      0.068
b19                   0.002 0.001      0.000    0.002      0.003
as.factor(Education)1 0.259 0.041      0.179    0.259      0.338
as.factor(Education)2 0.488 0.067      0.356    0.488      0.620
b4                    0.002 0.027     -0.051    0.002      0.054
v025                 -0.063 0.030     -0.122   -0.063     -0.005
BMI                  -0.004 0.003     -0.010   -0.004      0.002
```

```
par(mfrow = c(2, 2))  # display a unique layout for all graphs
## Posterior density plot for the intercepts
plot(model_linear$marginals.fixed[[1]], type = "l", main = "", ylab = "", xlab = expression(beta[0]))
plot(model_linear$marginals.fixed[[2]], type = "l", main = "", ylab = "", xlab = expression(beta[1]))
par(mfrow = c(1, 1))
```



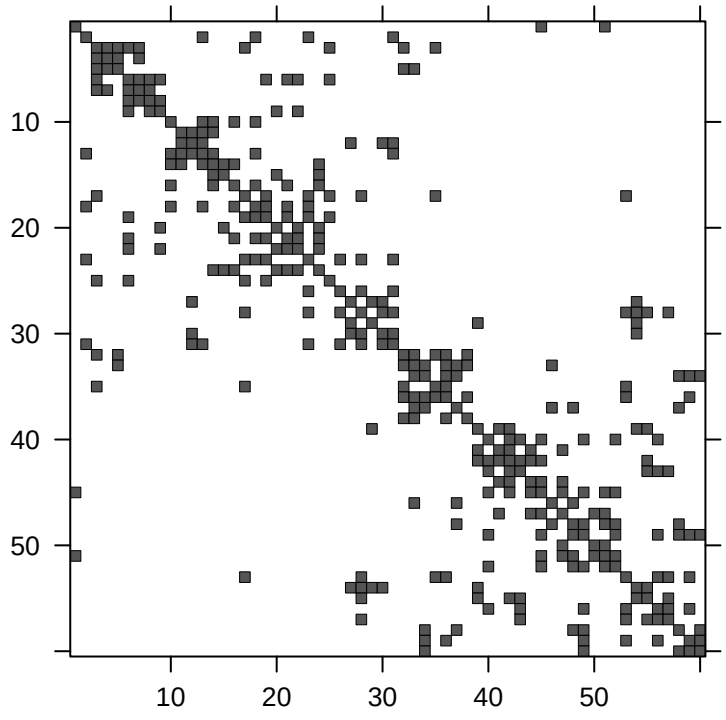

## Spatial Model in INLA

First we convert our neighbour map to an inla intergrated map , a map that INLA can use to evaluate its model

```
nb2INLA("zim_shape/zim_inla.graph", zim_nb)
zim_adj <- paste(getwd(), "/zim_shape/zim_inla.graph", sep = "")
```

Then we generate the adjacency matrix for the ZIM example: in the plot below rows and columns identify areas; squares identify neighbors

```
H <- inla.read.graph(filename = "zim_shape/zim_inla.graph")
image(inla.graph2matrix(H), xlab = "", ylab = "")
```




## Fitting CAR in INLA

After having defined our neighbourhood structure, then we need to specify the formula for the model, through

```
formula_inla <- Stunting ~ 1 + as.factor(Employed) +
  b19  + as.factor(Education) + b4+ v025+ BMI+

  ##our CAR specification
  f(id_2, model="bym",graph=zim_adj, scale.model=TRUE,
    ## spefiying the priors for the unstri and str
                    hyper=list(prec.unstruct=list(prior="loggamma",param=c(1,0.001)),
                               prec.spatial=list(prior="loggamma",param=c(1,0.001))))
```

where `id_2` represents the identifiers for the locations and through the graph option we include the name of the object containing the neighborhood structure. Note that with f(ID, model="bym",...) R-INLA parameterizes

$$\xi_i = u_i + v_i$$

By default, minimally informative priors are specified on the log of the unstructured effect precision $\log(\tau_v) \sim \log Gamma(1, 0.0005)$ and the log of the structured effect precision $\log(u_v) \sim \log Gamma(1, 0.0005)$

```
model_inla <- inla(formula_inla,family="gaussian",
                   data=zim_child_model,
                   control.compute=list(dic=TRUE))
```

```r
summary(model_inla)  #inla --> 17218.91(15.09) ##linear model--> 17272.83(9.444)
```

```
Call:
c("inla(formula = formula_inla, family = \"gaussian\", data = zim_child_model, ",  "    control.compute

Time used:
 Pre-processing    Running inla Post-processing           Total
         2.6105         27.2763          0.5672         30.4540

Fixed effects:
                          mean     sd 0.025quant 0.5quant 0.975quant
(Intercept)           -0.6439 0.1168    -0.8732  -0.6439    -0.4146
as.factor(Employed)1   0.0123 0.0279    -0.0424   0.0123     0.0670
b19                    0.0014 0.0008    -0.0001   0.0014     0.0030
as.factor(Education)1  0.2578 0.0412     0.1769   0.2578     0.3386
as.factor(Education)2  0.4842 0.0679     0.3508   0.4842     0.6175
b4                     0.0016 0.0268    -0.0510   0.0016     0.0540
v025                  -0.0588 0.0317    -0.1209  -0.0588     0.0033
BMI                   -0.0034 0.0031    -0.0094  -0.0034     0.0026
                          mode kld
(Intercept)           -0.6439   0
as.factor(Employed)1   0.0123   0
b19                    0.0014   0
as.factor(Education)1  0.2578   0
as.factor(Education)2  0.4842   0
b4                     0.0016   0
v025                  -0.0588   0
BMI                   -0.0034   0

Random effects:
Name      Model
 id_2   BYM model

Model hyperparameters:
                                          mean       sd 0.025quant
Precision for the Gaussian observations    0.932   0.0084     0.9151
Precision for id_2 (iid component)      3084.018 906.0567  1725.6866
Precision for id_2 (spatial component)   242.177 152.3191    45.1030
                                         0.5quant 0.975quant      mode
Precision for the Gaussian observations    0.9321     0.9483    0.9326
Precision for id_2 (iid component)      2942.1777  5248.7430 2679.2657
Precision for id_2 (spatial component)   211.5009   616.9183  129.7827

Expected number of effective parameters(std dev): 14.57(0.9388)
Number of equivalent replicates : 409.16

Deviance Information Criterion (DIC) ...............: 17218.97
Deviance Information Criterion (DIC, saturated) ....: 5846.34
Effective number of parameters .....................: 14.28

Marginal log-Likelihood:  -8637.43
Posterior marginals for linear predictor and fitted values computed
```

### Summary of the fixed effect

```
round(model_inla$summary.fixed, 3)
```

```
                        mean    sd 0.025quant 0.5quant 0.975quant   mode
(Intercept)           -0.644 0.117     -0.873   -0.644     -0.415 -0.644
as.factor(Employed)1   0.012 0.028     -0.042    0.012      0.067  0.012
b19                    0.001 0.001      0.000    0.001      0.003  0.001
as.factor(Education)1  0.258 0.041      0.177    0.258      0.339  0.258
as.factor(Education)2  0.484 0.068      0.351    0.484      0.617  0.484
b4                     0.002 0.027     -0.051    0.002      0.054  0.002
v025                  -0.059 0.032     -0.121   -0.059      0.003 -0.059
BMI                   -0.003 0.003     -0.009   -0.003      0.003 -0.003
                      kld
(Intercept)             0
as.factor(Employed)1    0
b19                     0
as.factor(Education)1   0
as.factor(Education)2   0
b4                      0
v025                    0
BMI                     0
```

### Summary of the random effects

```
round(head(model_inla$summary.random$id_2), 3)
```

```
  ID   mean    sd 0.025quant 0.5quant 0.975quant   mode kld
1  1 -0.234 0.045     -0.326   -0.233     -0.151 -0.230   0
2  2 -0.051 0.040     -0.134   -0.050      0.025 -0.048   0
3  3  0.046 0.057     -0.063    0.044      0.164  0.042   0
4  4  0.057 0.086     -0.107    0.054      0.241  0.050   0
5  5  0.050 0.075     -0.093    0.047      0.210  0.043   0
6  6  0.054 0.055     -0.053    0.053      0.167  0.052   0
```

### Plotting the estimates on map

FOr this excercise, we plot the probaility of having high HAZ in ZIM to compare with out BUGs model. FIrst extract the marginals of the random effects

```
csi <- model_inla$marginals.random$id_2[1:60]
```

Then create the posterior probablity using `inla.pmarginal` function.

```
# *** Code for posterior probablity
a <- 0
prob_csi <- lapply(csi, function(x) {
    1 - inla.pmarginal(a, x)
})
## for each location estimate the probability in continous
cont_prob_cs1 <- data.frame(maps_cont_prob_csi = unlist(prob_csi)) %>% tibble::rownames_to_column("ID_2
    mutate(ID_2 = gsub("index.", "", ID_2))

maps_cont_prob_csi <- cont_prob_cs1
```

Or create catoroes if you wish

```
# for each location estimate the probability in groups
prob_csi_cutoff <- c(0,0.2,0.4,0.6,0.8,1) ## can change accordingly
cat_prob_csi <- cut(unlist(prob_csi),
                    breaks=prob_csi_cutoff,
                    include.lowest=TRUE)


maps_cat_prob_csi <- data.frame(ID_2=unique(zim_child_model$id_2), ## check whether it joins well
                                cat_prob_csi=cat_prob_csi)
maps_cat_prob_csi$ID_2 <- as.character(maps_cat_prob_csi$ID_2)
```

Eventually, we jooin our posterior probabilities to the data

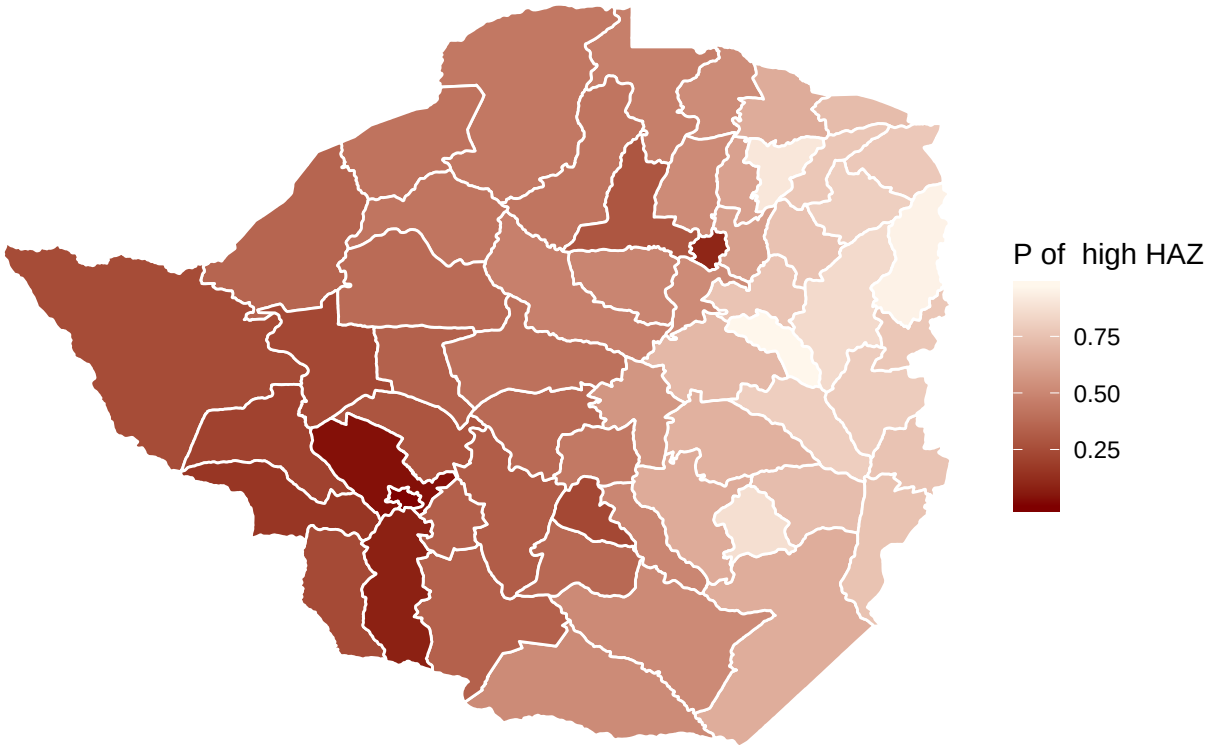
```
zim_shp_df <- broom::tidy(zim_shp, region = "ID_2")
zim_shp_df <- zim_shp_df %>%
 # left_join(maps_cat_prob_csi, by=c("id"="ID_2")) %>%
  left_join(maps_cont_prob_csi, by=c("id"="ID_2"))

glimpse(zim_shp_df)
```

```
Observations: 30,130
Variables: 8
$ long               <dbl> 28.61305, 28.60440, 28.57862, 28.55219, 28....
$ lat                <dbl> -20.23587, -20.22540, -20.22598, -20.22138,...
$ order              <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...
$ hole               <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, F...
$ piece              <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ group              <chr> "1.1", "1.1", "1.1", "1.1", "1.1", "1.1", "...
$ id                 <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1"...
$ maps_cont_prob_csi <dbl> 3.088706e-09, 3.088706e-09, 3.088706e-09, 3...
```

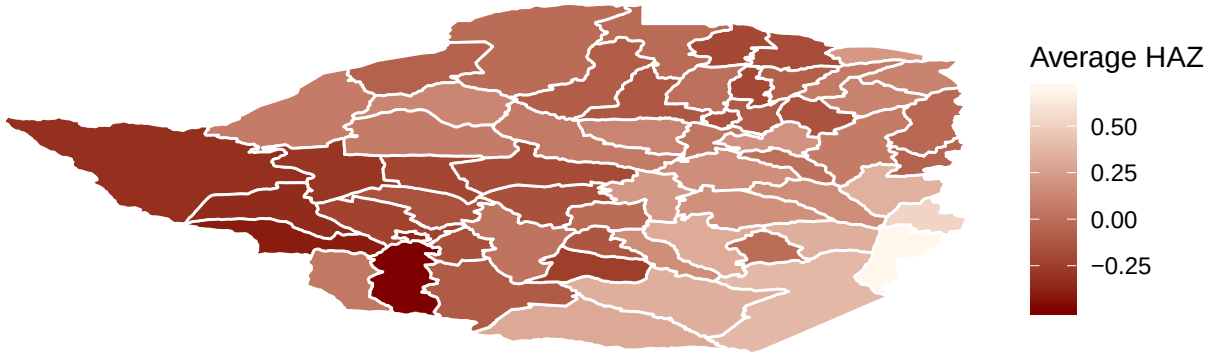
We have our plot, is it similar

```
p2 <- ggplot() + geom_polygon(data = zim_shp_df, aes(x = long, y = lat, group = group,
    fill = maps_cont_prob_csi), colour = "white") + theme_void() + ggtitle("RINLA Fit") +
    labs(fill = "P of  high HAZ") + scale_fill_continuous(high = "#fff7ec",
    low = "#7F0000")
p2
```

RINLA Fit



Open Bugs Fit




RINLA Fit