```
                    OHIO LUNG CANCER RISK ANALYSIS


library(R2jags)
library(MCMCvis)
library(coda)
library(lattice)
library(tidyverse)
library(maps)
library(ggplot2)
library(caret)
library(class)
library(ggplot2)
library(kernlab)
library(tidyverse)
library(knitr)
library(naniar)
library(randomForest)
library(C50)
library(MASS)
library(e1071)
data_ohio = read.csv("Ohio_Data.csv")


df = summary(data_ohio)
kable(df,"simple", caption = "Ohio data summary")
```

Ohio data summary

X	Obs	Exp
Min. : 1.00	Min. : 4.00	Min. : 6.902
1st Qu.:22.75	1st Qu.: 15.75	1st Qu.: 17.754
Median :44.50	Median : 26.00	Median : 28.204
Mean :44.50	Mean : 69.45	Mean : 69.454
3rd Qu.:66.25	3rd Qu.: 57.00	3rd Qu.: 53.470
Max. :88.00	Max. :922.00	Max. :991.174

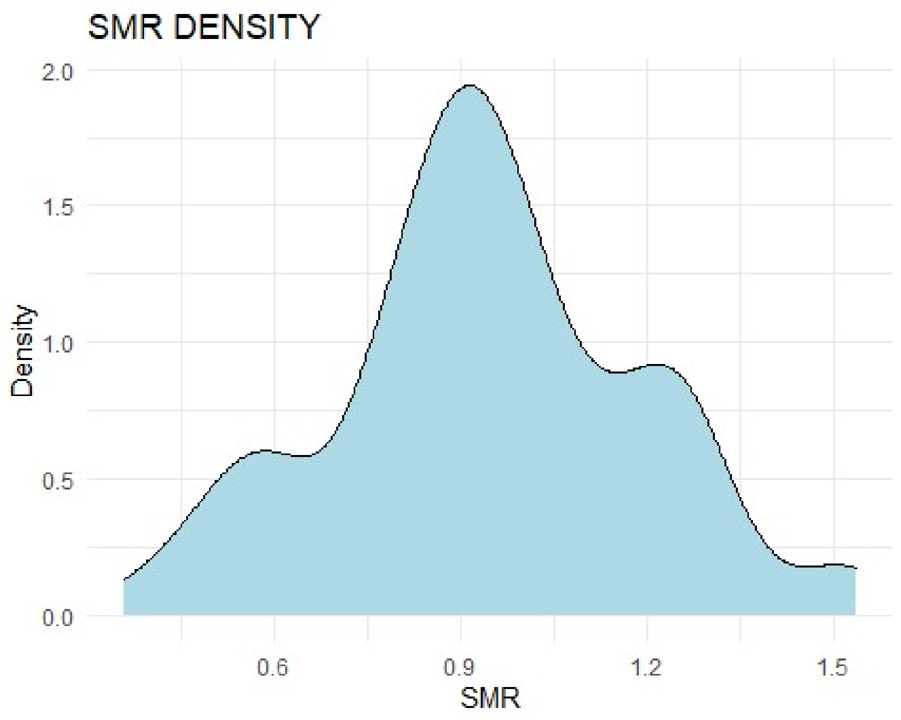
In the dataset Ohio_Data, we observe the number of lung cancer and the expected number of lung cancer for 88 counties in Ohio in 1988. It is observed from the summary that some counties have large observed and expected lung cancer counts indicating uneven number of cases in each county. This may be due to difference in population of each county. The data set does not have any missing values.

```
df = data.frame(SMR = data_ohio$Obs/data_ohio$Exp)
kable(summary(df))
```

SMR
Min. :0.3599
1st Qu.:0.8109
Median :0.9293
Mean :0.9433
3rd Qu.:1.0834
Max. :1.5369

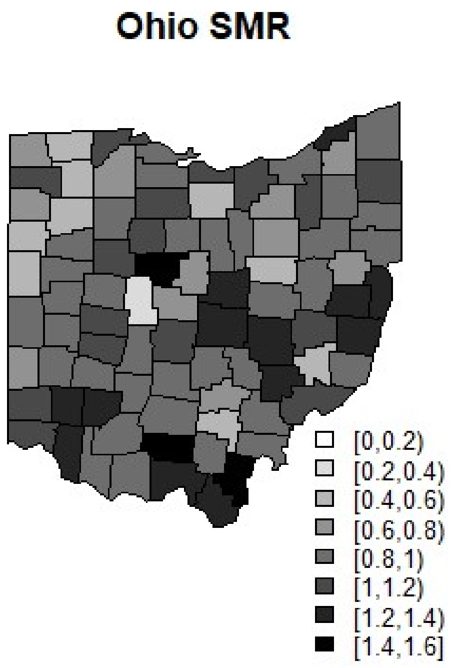
The observed number of death or disease can be compared to the expected number using Standard mortality rate (SMR). We compare this by dividing the observed number of diseases with the expected number of diseases.

```
# DISTRIBUTION OF SMR
ggplot(df) +
 aes(x = SMR) +
 geom_density(adjust = 1L, fill = "lightblue") +
  labs(title = 'SMR DENSITY', y = 'Density') +
 theme_minimal()
```



From the density plot we observe that most of the SMR is bellow 1. This means we usually observe less lung cancer than we expect.

```r
# MAP OF SMR
source("OhioMap.R") # need to read in the OhioMap function
OhioMap(df$SMR,ncol=8,type="e",figmain="Ohio SMR",lower=0,upper=1.6)
```



From the plot we observe that the county Marion,Pike and Meigs have high SMR, these counties have almost 1.4 times more lung cancer than expected. Most of the other counties observe lower SMR with the lowest seen at Union county.

When building a model for the observed value we offset it on the expected number of lung cancer. This is because each county has a large variation in the number of lung cancer cases from each other. This may be due to a difference in population. To directly compare these counties we are providing an offset. For this, we adjust the mean of the Poisson distribution. Instead of modelling the count of observed lung cancer, we model the mean rate of occurrence by using the product of a known quantity Exp(expected count) and rate theta.

$$\mu = Exp * \rho$$

$$log(\mu) = log(Exp) + log(\rho)$$

$$log(\rho) = \beta_o + \beta_i(x_i)$$

$$log(\theta_i) = \beta_i(x_i)$$

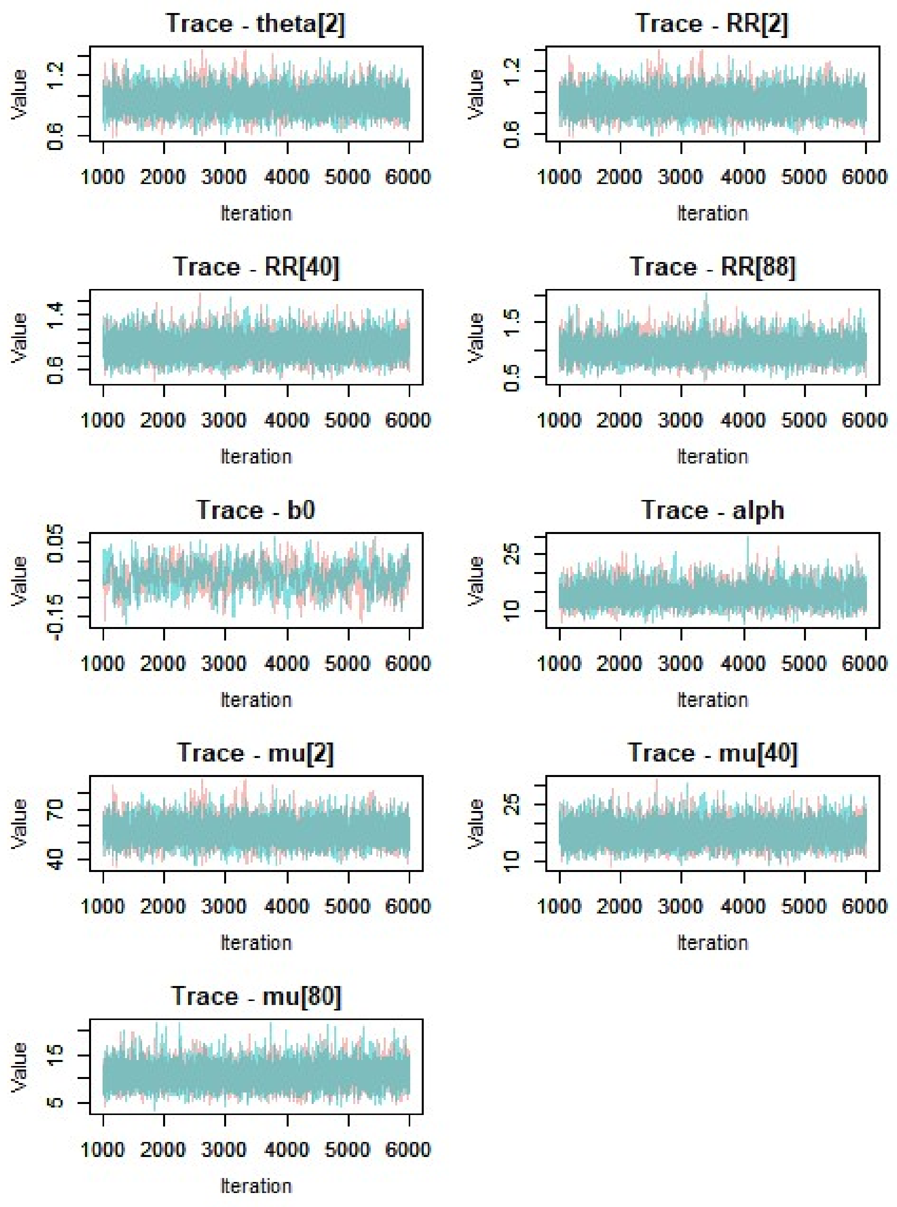
Theta gives us explanatory variables and their covariates and Beta Zero is the intercept. Relative risk RR is used to compare events occurring between groups. Here the RR for each county can be calculated from an equation with the explanatory variable and its covariates (theta). It changes directly with a change in the explanatory variable and its covariates (theta) and changes exponentialy with intercept (beta zero).

```r
#FUNCTION :
ohio_Mortality <- function(){
  b0 ~ dunif(-100,100) # prior
  alph ~ dgamma(1,1) # prior
  for(i in 1:N){
  theta[i] ~ dgamma(alph,alph)
  Obs[i] ~ dpois(mu[i]) # likelihood
  log(mu[i]) <- log(Exp[i])+ b0 + log(theta[i]) # prediction
  RR[i] <- exp(b0)*theta[i]
  }
}

# Data:
Exp = data_ohio$Exp
Obs = data_ohio$Obs
N = 88

ohio_model_data = list('Exp','Obs','N')

# INITIAL VALUES
inits1 <- list( 'alph' =0.9, 'b0' = 0.005) # chain 1
inits2 <- list('alph' = 1.1, 'b0' = 0.004) # chain 2
int_ohio_par <- list(inits1,inits2)

#PARAMETERS OF INTREST
param <- c('RR','alph','b0','mu','theta')


# MODEL
ohio_model  <-  jags(data = ohio_model_data,
                inits = int_ohio_par,
                n.iter = 12000,
                parameters.to.save = param,
                model.file = ohio_Mortality,
                n.burnin = 6000,
                n.chains=2,n.thin=1)

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 88
##    Unobserved stochastic nodes: 90
##    Total graph size: 711
##
## Initializing model
```

We build a jags model for the Ohio dataset to predict the observed value and relative risk for each county. We use a flat prior or uniform prior for Beta zero which means that the MLE coincides with the maximum posterior probability and the prior has no effect on likelihood. The prior alph comes from a Gamma(1,1) distribution which is exponential with a mean of 1. Then we have the likelihood obs[i] to which we pass the observed data. log(mu[i]) is the regression model with offset. The Relative risk RR depends on theta. Here log in log(mu) is a link function used to connect the positive mean observed count with the regression model which is linear in the betas.

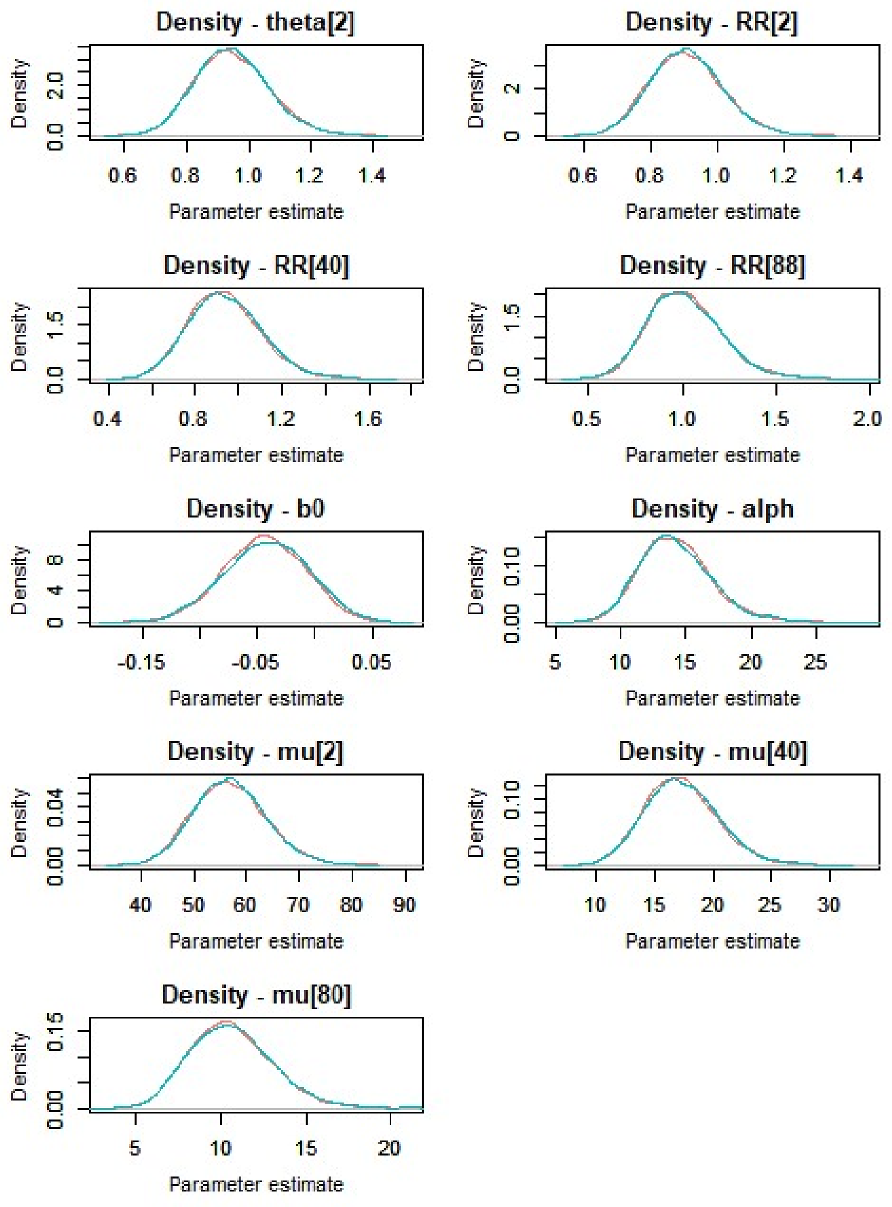
To monitor convergence we are using 2 chains. If the chains can't be distinguished then we assume convergence. For this purpose, we have two initial values. The initial values for the priors 'Alph' and 'beta 0' are provided based on likely values in the prior distribution. Then we create a list of all parameters we want to monitor. 'Obs' even though is a stochastic node it is not a parameter we want to monitor as it is the data itself. We monitor prior even though it comes from known given distribution as it will be updated by the likelihood to a posterior. We then build a model which generates 12000 samples and remove the first 6000 samples as burn-in.

```
#TRACE PLOT:
mcmc_ohio <- as.mcmc(ohio_model)

MCMCtrace(mcmc_ohio,
          params = c('theta\\[2\\]','RR\\[2\\]','RR\\[40\\]','RR\\[8
8\\]',
                     'b0','alph','mu\\[2\\]','mu\\[40\\]','mu\\[80\\
]'),#param intrest
          type = 'trace', # trace plot
          ind = TRUE,     # separate density lines for each chain
          pdf = FALSE, ISB = FALSE,
          exact=FALSE)
```



```
MCMCtrace(mcmc_ohio,
          params = c('theta\\[2\\]','RR\\[2\\]','RR\\[40\\]','RR\\[8
8\\]',
                     'b0','alph','mu\\[2\\]','mu\\[40\\]','mu\\[80\\
]'),#param intrest
          type = 'density', # trace plot
          ind = TRUE,      # separate density lines for each chain
```

```
           pdf = FALSE, ISB = FALSE,
           exact=FALSE)
```



From the trace plot, we observe that the two chains have similar behavior for all parameters even though different initial points are provided therefore we can assume convergence.

```
##      Point est. Upper C.I.
## alph   1.000344   1.001235
## b0     1.003115   1.006609
```

```
##        Point est. Upper C.I.
## mu[1]   1.0008467   1.000899
## mu[18]  0.9999893   1.000278
```

```
##        Point est. Upper C.I.
## RR[1]   1.0008467  1.0008995
## RR[18]  0.9999893  1.0002780
## RR[63]  0.9999238  0.9999484
```

```
##           Point est. Upper C.I.
## theta[35]   1.000828   1.004068
## theta[8]    1.000000   1.000006
## theta[79]   1.000190   1.000768
```

To make sure that the parameters converge we look at 'Gelman.diag' it gives us the upper limit for scale reduction. This is calculated from within and between variance of the two chains. The upper C. I should be less than 1.1 and here from the table, we observe the same. All parameters have upper C. I less than 1.1. Looking at both the trace plots and 'gelmans.diag' we deduce that the parameters are converged. Now we can look at the summary of the model.

```
##             mean        sd      2.5%       25%       50%       75%
97.5%
## RR[1] 0.9242481 0.1747660 0.6160772 0.8001987 0.9129617 1.0373906
1.302246
## RR[2] 0.9063108 0.1094268 0.7049924 0.8303990 0.9017727 0.9761872
1.132993
## RR[3] 0.9644833 0.1550552 0.6878135 0.8554814 0.9571696 1.0653777
1.285946
## RR[4] 0.9593395 0.1130337 0.7531121 0.8804086 0.9548145 1.0320775
1.193077
## RR[5] 0.8686768 0.1461997 0.6120175 0.7665414 0.8603633 0.9597699
1.180180
##           Rhat n.eff
## RR[1] 1.000997 12000
## RR[2] 1.000959 12000
## RR[3] 1.000915 12000
## RR[4] 1.001552  4000
## RR[5] 1.001764  1800
```

```
##             mean         sd       2.5%         25%         50%
75%
## alph 14.19211263 2.71469023  9.4870324 12.29215663 13.96910792 15
.87388857
## b0   -0.04251699 0.03602433 -0.1141883 -0.06687309 -0.04187872 -0
```

```
.01703704
##           97.5%     Rhat n.eff
## alph 20.1300791 1.001234  4700
## b0    0.0260062 1.003115  1700
```

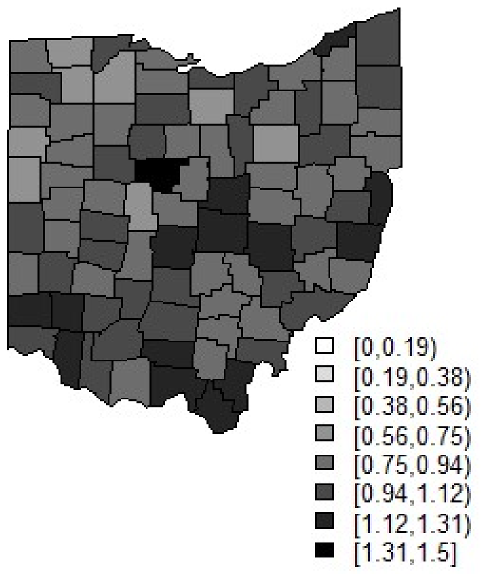
```
##            mean       sd      2.5%      25%      50%      75%    9
7.5%     Rhat
## mu[1] 14.49069 2.740044  9.659078 12.54580 14.31374 16.26458 20.4
1708 1.000997
## mu[2] 56.90406 6.870525 44.263991 52.13783 56.61914 61.29136 71.1
3662 1.000959
## mu[3] 25.99609 4.179262 18.538900 23.05812 25.79896 28.71553 34.6
6059 1.000915
##       n.eff
## mu[1] 12000
## mu[2] 12000
## mu[3] 12000
```

```
##                mean        sd      2.5%        25%        50%      7
5%    97.5%
## theta[1] 0.9644726 0.1829610 0.6393875 0.8341977 0.9526129 1.0801
58 1.358220
## theta[2] 0.9461022 0.1177174 0.7341085 0.8641553 0.9407654 1.0215
83 1.191034
## theta[3] 1.0066379 0.1636452 0.7137900 0.8915434 0.9988293 1.1117
09 1.344535
##              Rhat n.eff
## theta[1] 1.001252 12000
## theta[2] 1.000919 12000
## theta[3] 1.000950 12000
```

From the summary of the model, we get the estimated Relative risk and predicted mean of observed lung cancer. We also get the point estimate of the mean of the distribution for all the parameters (mean in the table) is extracted. We also get the standard deviation along with quantiles. From the summary, we can easily get the credible interval for each parameter. The credible interval for theta does not contain any zero which shows that it has a relavent influence on the mean with probability 95%(CI 2.5 to 97.25).

```r
# POSTERIOR RR PLOT
RR_posterior = ohio_model$BUGSoutput$summary[1:88,1]

source("OhioMap.R")
# need to read in the OhioMap function
OhioMap(RR_posterior,ncol=8,type="e",figmain="Ohio Relative Risk Pos
terior Plot",lower=0,upper=1.5)
```

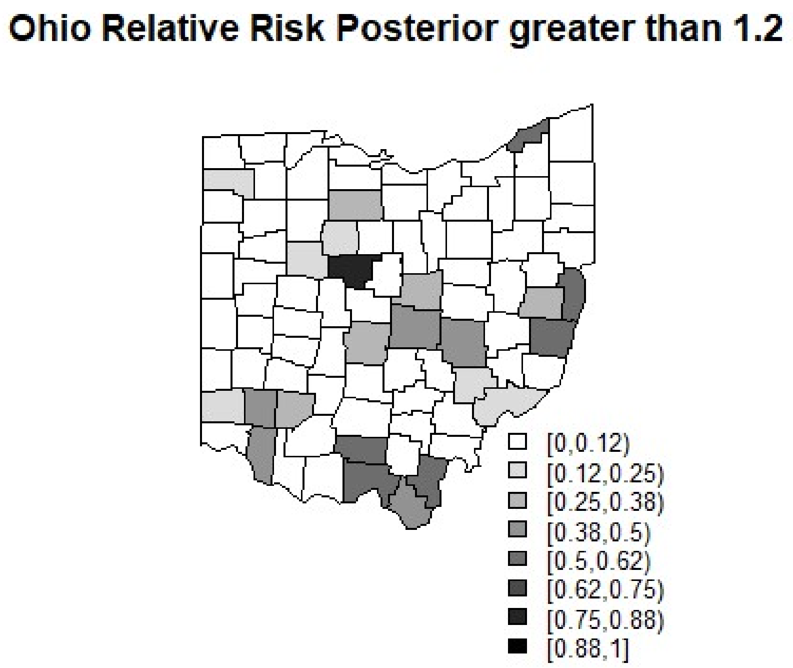
Ohio Relative Risk Posterior Plot



From the figure we see that regions with highest relative risks is Marion. we also see other counties with great relative risk shaded in dark color.

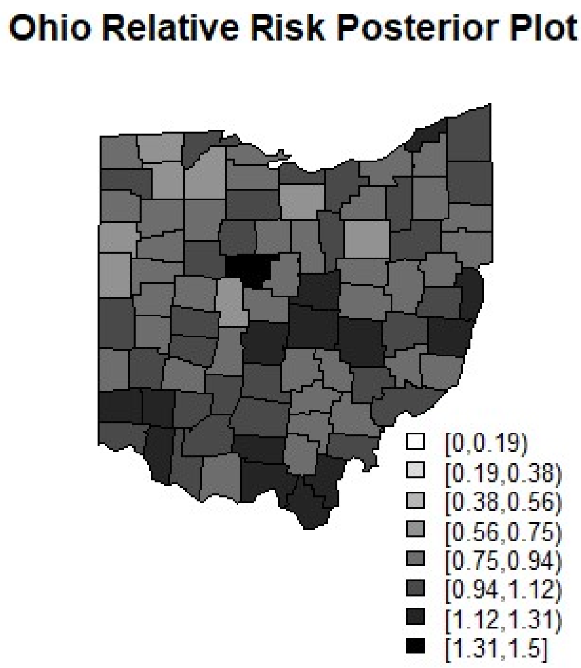
```
# POSTERIOR PROB OF RR> 1.2
ohio_Mortality <- function(){
  b0 ~ dunif(-100,100)
  alph ~ dgamma(1,1)
  for(i in 1:N){
    Obs[i] ~ dpois(mu[i])
    log(mu[i]) <- log(Exp[i])+ b0 + log(theta[i])
    theta[i] ~ dgamma(alph,alph)
    RR[i] <- exp(b0)*theta[i]
    P.RR[i] <- ifelse(RR[i] > 1.2 ,1,0) # probability of interest
  }
}

param <- c('P.RR')

ohio_model_prob    <- jags(data = ohio_model_data,
                      inits = int_ohio_par,
                      n.iter = 12000,
                      parameters.to.save = param,
                      model.file = ohio_Mortality,
                      n.burnin = 6000,
                      n.chains=2,n.thin=1)

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 88
##    Unobserved stochastic nodes: 90
##    Total graph size: 889
##
## Initializing model
```

```
#print(ohio_model)

#PROB RR> 1.2 PLOT
P.RR_posterior = ohio_model_prob$BUGSoutput$summary[1:88,1]

source("OhioMap.R")
testdat <- runif(88) # need to read in the OhioMap function
OhioMap(P.RR_posterior,ncol=8,type="e",figmain="Ohio Relative Risk P
osterior greater than 1.2",lower=0,upper=1)
```

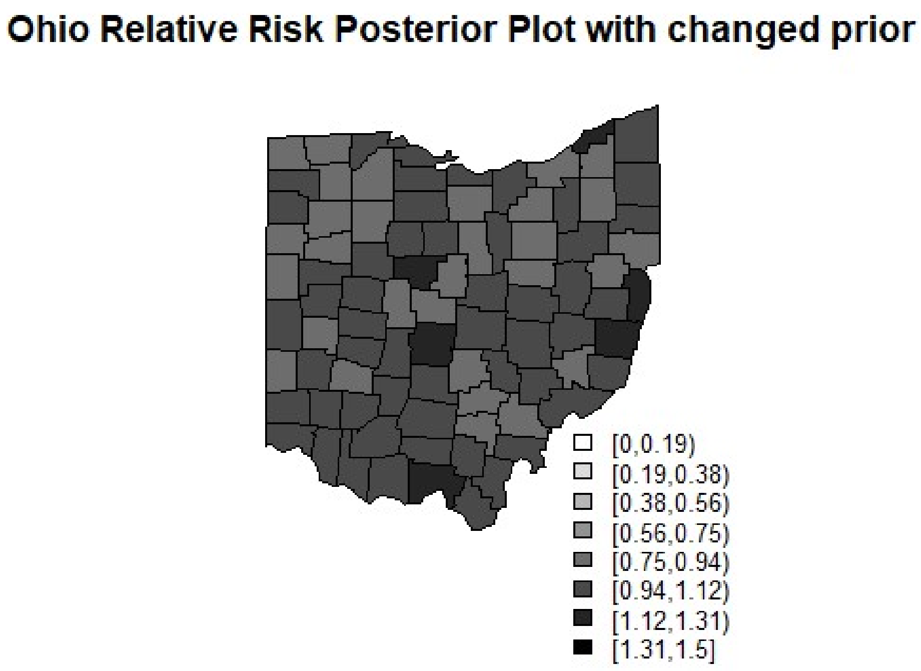


We want to find the probability that relative risk in each region exceed 1.2. For this We modify the model to account for the probability of interest. When we map it we see regions with high probability of relative risk greater than 1.2. The county Marion has the highest probability with 88 percent probability that the county has relative risk greater than 1.2.

```
ohio_Mortality_pd <- function(){
  b0 ~ dnorm(-0.04,0.01) # informative prior:
  alph ~ dgamma(0.001,0.001) # vauge prior
  for(i in 1:N){
    Obs[i] ~ dpois(mu[i])
    log(mu[i]) <- log(Exp[i])+ b0 + log(theta[i])
    theta[i] ~ dgamma(alph,alph)
    RR[i] <- exp(b0)*theta[i]
    P.RR[i] <- ifelse(RR[i] > 1.2 ,1,0)
  }
}


# Data:
Exp = data_ohio$Exp
Obs = data_ohio$Obs
N = 88

ohio_model_data = list('Exp','Obs','N')

# INITIAL VALUES
inits1 <- list( 'alph' =0.9, 'b0' = 0.005) # chain 1
inits2 <- list('alph' = 1.1, 'b0' = 0.004) # chain 2
int_ohio_par <- list(inits1,inits2)

#PARAMETERS OF INTREST
param <- c('RR','alph','b0','mu')


# MODEL
ohio_model_pd  <-  jags(data = ohio_model_data,
                inits = int_ohio_par,
                n.iter = 10000,
                parameters.to.save = param,
                model.file = ohio_Mortality_pd,
                n.burnin = 500,
                n.chains=2,n.thin=1)

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 88
##    Unobserved stochastic nodes: 90
##    Total graph size: 891
##
## Initializing model

# PRIORS CHANGED
ohio_model_pd$BUGSoutput$summary[89:90,]

##             mean          sd        2.5%         25%         50%
75%
## alph 60.35277872 28.33238508 26.57155109 41.29393510 53.59850745
71.442783391
## b0   -0.02417667  0.02310928 -0.07060884 -0.03946405 -0.02352748
-0.008558426
##             97.5%     Rhat n.eff
## alph 135.37114863 1.006298   530
## b0     0.01961933 1.000949 19000

#SAME PRIOR
ohio_model$BUGSoutput$summary[89:90,]

##              mean          sd       2.5%          25%          50%
75%
## alph 14.19211263 2.71469023  9.4870324 12.29215663 13.96910792 15
.87388857
## b0   -0.04251699 0.03602433 -0.1141883 -0.06687309 -0.04187872 -0
.01703704
##           97.5%     Rhat n.eff
## alph 20.1300791 1.001234  4700
## b0    0.0260062 1.003115  1700

# PLOTING THE RR FOR DIFFRENT PRIORS

RR_posterior_pd = ohio_model_pd$BUGSoutput$summary[1:88,1]

source("OhioMap.R")
# need to read in the OhioMap function
#map.text("county","ohio")
OhioMap(RR_posterior,ncol=8,type="e",figmain="Ohio Relative Risk Pos
terior Plot",lower=0,upper=1.5)
```



```
OhioMap(RR_posterior_pd,ncol=8,type="e",figmain="Ohio Relative Risk
Posterior Plot with changed prior",lower=0,upper=1.5)
```



To understand the effect of prior we provide ‘beta zero’ as an informative prior and ‘alph’ as a vague prior. In the original model, the prior ‘beta zero’ was a vauge prior and ‘alph’ was an informative prior. When the prior is vague the posterior would be proportional to the likelihood, therefore prior does not have any influence on the likelihood. This means when the prior is vague the posterior depends on the data and the prior provides equal weight for all data points.

From the plot, we can see that when the priors are changed the Relative Risk (RR) map changes. We observe that ‘RR’ for each county has increased and almost uniform ‘RR’ is observed for all counties in Ohio than previously. This is because RR comes from theta which comes from the vague prior ‘alph’ which we changed to, As ‘alph’ is vague all the ‘RR’ for most of the county has equal weight and we observed almost uniform ‘RR’ in Ohio counties. As ‘beta zero’ was changed to informative prior with higher mean point estimate the RR increased in all counties. Therefore we can see that priors have a great effect on the model.

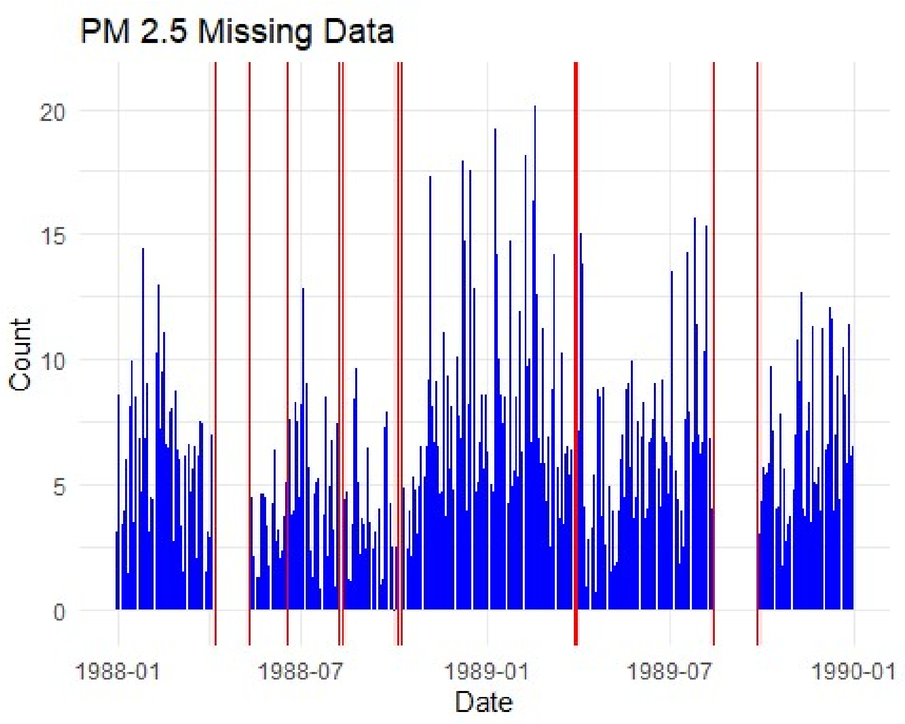
```
# DATA ANALYSIS:
data_ohio_pm25 = read.csv("ohio_pm25.csv")
summary(data_ohio_pm25) # 95 NA and a negative value for pm2.5 prese
nt is present in row 243,276

##      Date                pm2.5
##  Length:731         Min.   :-0.400
##  Class :character   1st Qu.: 3.746
##  Mode  :character   Median : 5.600
##                     Mean   : 6.269
##                     3rd Qu.: 8.200
##                      Max.   :20.800
##                      NA's   :95
```

```
miss_var_summary(data_ohio_pm25)
```

```
## # A tibble: 2 × 3
##   variable n_miss pct_miss
##   <chr>     <int>    <dbl>
## 1 pm2.5        95     13.0
## 2 Date          0        0
```

From the initial dataset analysis, it is observed that the dataset contains missing data. About 12.9% of the data is missing (95 data points). It is also observed that PM 2.5 particulate matter which is supposed to be positive has negative data points in some regions with the lowest recorded as -0.4.

```
# plot with regions of missing para

ggplot(data_ohio_pm25) +
 aes(x = as.Date(Date), weight = pm2.5) +
 geom_bar(fill = "blue") +
  labs(title = 'PM 2.5 Missing Data', y = 'Count', x = 'Date')+
 theme_minimal()+
  geom_vline(xintercept=c(as.Date('1988-04-06'),as.Date('1988-05-11'
)), colour = 'red')+
  geom_vline(xintercept=c(as.Date('1989-08-14'),as.Date('1989-09-27'
)), colour = 'red')+
  geom_vline(xintercept= as.Date('1988-06-18'), colour = 'red')+
  geom_vline(xintercept=c(as.Date('1988-08-08'),as.Date('1988-08-12'
)), colour = 'red')+
  geom_vline(xintercept=c(as.Date('1988-10-05'),as.Date('1988-10-09'
)), colour = 'red')+
  geom_vline(xintercept=c(as.Date('1989-03-29'),as.Date('1989-04-01'
)), colour = 'red')
```



The red lines in the plot illustrate the period of missing data.

```
# ANALYSISING OHIO DATA FOR 1988:


# MODEL
jags.mod.ohio <- function(){
  # Observation model
  for (i in 2 : N) {
    Ohio[i] ~ dnorm(Y[i],tau.v)
  }
  Ohio[1] ~ dnorm(Y[1],tau.v)
  tau.v ~ dgamma(1,0.01)
  # System model
  for(i in 2:N){
    Y[i] ~ dnorm(Y[i-1],tau.w)
  }
  Y[1] ~ dnorm(6,0.001)
  tau.w ~ dgamma(1,0.01)
  sigma.w <- 1/sqrt(tau.w)
}
```

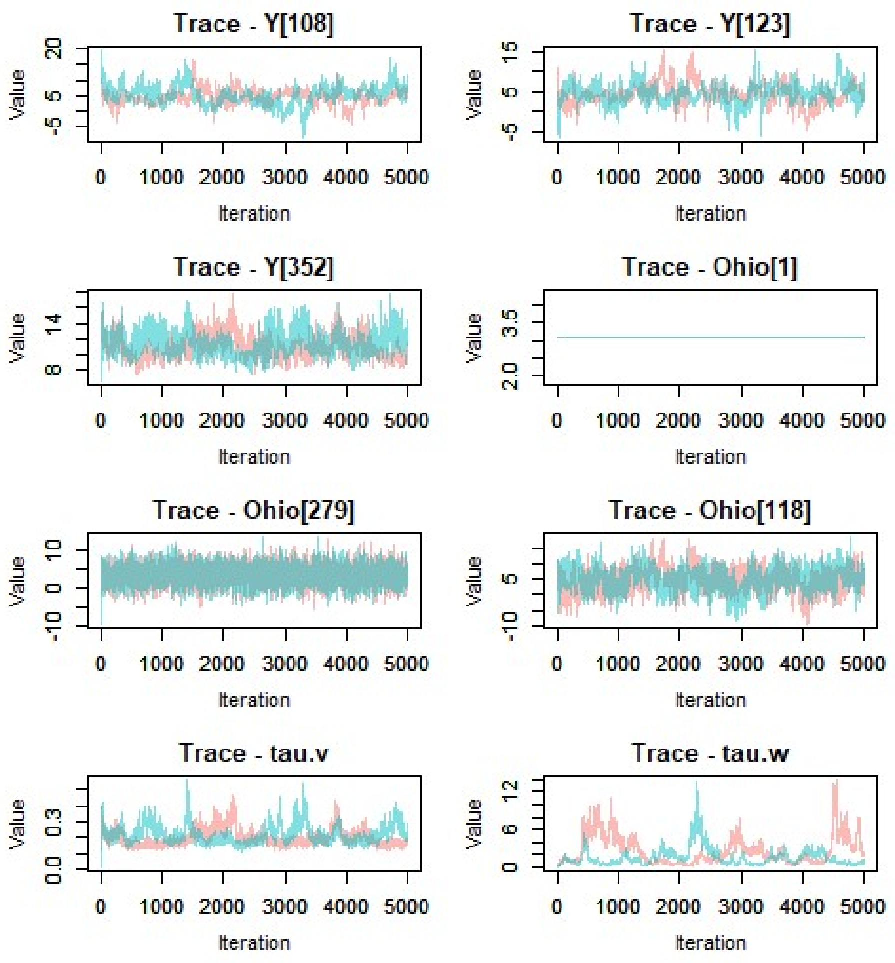
The value of measured PM 2.5 for each day is a normal distribution with mean as the true value (Y) and precision tau.v indicating error in measurement. The true PM2.5 value (Y) for each day is correlated to the true value of PM2.5 of the previous day with some random precision tau.w which we consider as white noise.

```
#DATA:
Ohio = data_ohio_pm25[1:366,]$pm2.5
```

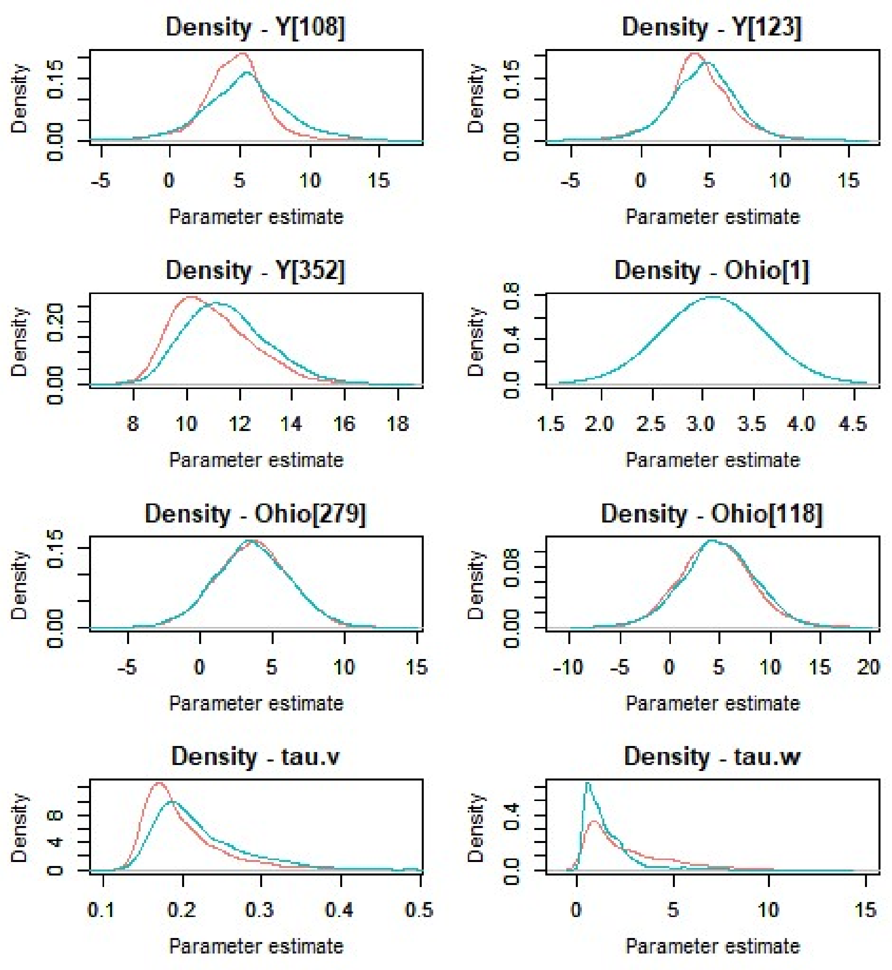
```
N = 366
jags.data.ohio.pm <- list("Ohio","N")


# Initial values:
set.seed(123)
initial_chain1 <- list(tau.v = 0.08, Y= runif(366,5.95,6.05),
               tau.w = 0.049,Ohio = c(rep(NA,96),rep(6.269,36),
                                      rep(NA,37),6.269,rep(NA,50),
                                      rep(6.269,5),rep(NA,53),6.269,
NA,
                                      rep(6.269,3),rep(NA,83))) # ch
ain 1  # explain na

initial_chain2 <- list(tau.v = 0.004, Y= runif(366,5.95,6.05),
               tau.w = 0.01,Ohio = c(rep(NA,96),rep(5.6,36),
                                      rep(NA,37),5.6,rep(NA,50),
                                      rep(5.6,5),rep(NA,53),5.6,NA,
                                      rep(5.6,3),rep(NA,83))) # chai
n 2

inital_values = list(initial_chain1,initial_chain2)

# Param to save:

ohio_param_save = c("Ohio","Y","tau.v","tau.w")

# modeling:

jags.mod.fit <- jags(data = jags.data.ohio.pm, inits = inital_values
,
                     parameters.to.save = ohio_param_save, n.chains
= 2,
                     n.iter = 10000,n.burnin = 5000,
                     n.thin=1,model.file = jags.mod.ohio )

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 320
##    Unobserved stochastic nodes: 414
##    Total graph size: 741
##
## Initializing model
```

After building the model we provide initial values. As there are missing data points we need to provide initial values for these to predict them. For this, we initialize missing data in two chains with a mean(6.269) and median(5.6). We use two chains to see if they are converging properly. If the two chains have different behaviour they are not converged. We also provide initial values for the stochastic nodes ‘tau. v’, ‘tau.w’ and ‘Y’. We also provide n.itter as 10,000 and n.burnin as 5,000. We provide n.burnin to give Markov chain time to reach its equilibrium distribution.

```
mcmc_ohio_pm2.5 = as.mcmc(jags.mod.fit)

MCMCtrace(mcmc_ohio_pm2.5,
          params = c('Y\\[108\\]','Y\\[123\\]','Y\\[352\\]',
                  'Ohio\\[1\\]','Ohio\\[1\\]','Ohio\\[279\\]',
                  'Ohio\\[118\\]','tau.v','tau.w'), # interest param
eters
          type = 'trace', # trace plot
          ind = TRUE, # separate density lines for each chain
          pdf = FALSE, ISB = FALSE,
          exact=FALSE)
```



```
MCMCtrace(mcmc_ohio_pm2.5,
          params = c('Y\\[108\\]','Y\\[123\\]','Y\\[352\\]','Ohio\\[
1\\]',
                     'Ohio\\[1\\]','Ohio\\[279\\]','Ohio\\[118\\]',
                     'tau.v','tau.w'), # interest parameters
          type = 'density', # trace plot
          ind = TRUE, # separate density lines for each chain
          pdf = FALSE, ISB = FALSE,
          exact=FALSE)
```



```
#GELMAN
gelman_df2 = gelman.diag(mcmc_ohio_pm2.5,multivariate = F)
gelman_df2$psrf[c(2,113),]
```

```
##          Point est. Upper C.I.
## Ohio[1]        NaN        NaN
## Ohio[2]        NaN        NaN
```

```
gelman_df2$psrf[c(23,24),]
```

```
##            Point est. Upper C.I.
## Ohio[118]   1.002007   1.010368
## Ohio[119]   1.004153   1.020134
```

```
gelman_df2$psrf[200:201,]
```

```
##            Point est. Upper C.I.
## Ohio[278]        NaN        NaN
## Ohio[279]   1.000057    1.00011
```

```
gelman_df2$psrf[380:381,]
```

```
##         Point est. Upper C.I.
## Y[108]   1.072548   1.210969
## Y[109]   1.067470   1.198578
```

```
gelman_df2$psrf[651:652,]
```

```
##         Point est. Upper C.I.
## Y[352]   1.060384   1.245838
## Y[353]   1.059118   1.241945
```

```
gelman_df2$psrf[734:735,]
```

```
##        Point est. Upper C.I.
## Y[98]   1.048105   1.133657
## Y[99]   1.055006   1.147272
```

```
gelman_df2$psrf[368:369,]
```

```
##        Point est. Upper C.I.
## tau.v   1.076938   1.290121
## tau.w   1.205970   1.743573
```

```
jags.mod.fit$BUGSoutput$summary[1:2,]
```

```
##          mean sd 2.5% 25% 50% 75% 97.5% Rhat n.eff
## Ohio[1]   3.1  0  3.1 3.1 3.1 3.1   3.1    1     1
## Ohio[2]   3.0  0  3.0 3.0 3.0 3.0   3.0    1     1
```

```
jags.mod.fit$BUGSoutput$summary[118:119,] # missing data
```

```
##                mean       sd      2.5%      25%      50%      75%
97.5%
## Ohio[118] 4.722675 3.576378 -2.324278 2.406399 4.693315 7.093907
11.78666
## Ohio[119] 4.636982 3.567495 -2.421706 2.311984 4.633171 6.935433
11.87848
```

```
##                 Rhat n.eff
## Ohio[118] 1.002989   720
## Ohio[119] 1.004910   380
```

```
jags.mod.fit$BUGSoutput$summary[278:279,] # missing data
```

```
##                 mean       sd      2.5%      25%      50%      75%
97.5%
## Ohio[278] 2.500000 0.000000  2.500000 2.500000 2.500000 2.500000
2.500000
## Ohio[279] 3.631126 2.517367 -1.332183 1.938944 3.624459 5.321053
8.550986
##                 Rhat n.eff
## Ohio[278] 1.000000     1
## Ohio[279] 1.000912 10000
```

```
jags.mod.fit$BUGSoutput$summary[474:475,]
```

```
##              mean       sd       2.5%      25%      50%      75%
97.5%
## Y[108] 5.023954 2.748274 -0.6043276 3.427897 5.013254 6.446638 11
.12632
## Y[109] 5.011044 2.767154 -0.5376642 3.394669 4.980355 6.513546 11
.11808
##              Rhat n.eff
## Y[108] 1.072548    51
## Y[109] 1.067470    53
```

```
jags.mod.fit$BUGSoutput$summary[718:719,]
```

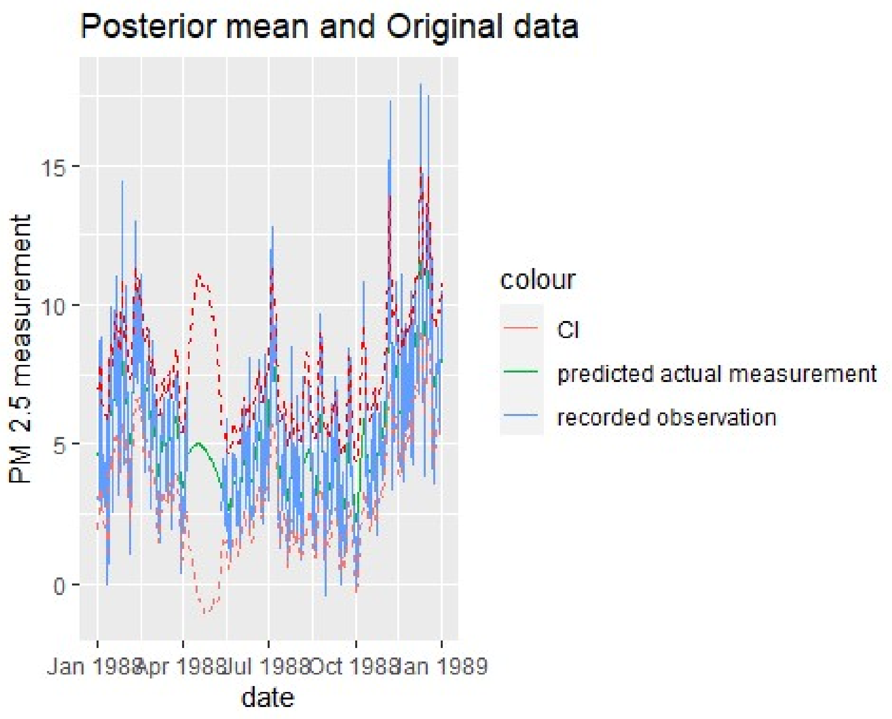
```
##              mean       sd     2.5%       25%      50%      75%
97.5%     Rhat
## Y[352] 11.23533 1.521385 8.773551 10.092628 11.07778 12.22032 14.
53536 1.062245
## Y[353] 11.02882 1.413249 8.650607  9.967356 10.91515 11.98004 13.
98637 1.061229
##        n.eff
## Y[352]    30
## Y[353]    30
```

```
jags.mod.fit$BUGSoutput$summary[734:735,]
```

```
##              mean         sd      2.5%       25%       50%       75
%     97.5%
## tau.v 0.2106853 0.05656546 0.1432787 0.1715789 0.1954379 0.234621
8 0.3581791
## tau.w 2.0504219 1.89328208 0.3406874 0.7663376 1.3725686 2.621883
4 7.3879462
##            Rhat n.eff
## tau.v 1.080186    25
## tau.w 1.163920    14
```

From the trace plot, we observe the following. Most of the predicted 'Y' do not converge. Ohio[1] is a straight line as there is no missing value and is the value we input in the dataset therefore there is no deviance. Ohio[279] converges but ohio[118] does not converge well. This might be the effect of prior. Here 'Y' has very low precision it might be one of the reasons for not converging. Better priors might fix the non-convergence. Another way to fix is by increasing the n.burnin and n.iter. Increasing n.burnin will give more time for the Markov chain to reach a steady state.

```
Y_posterior_mu = jags.mod.fit$BUGSoutput$summary[367:732,1]
CI_2.5 = jags.mod.fit$BUGSoutput$summary[367:732,3]
CI_97.5 = jags.mod.fit$BUGSoutput$summary[367:732,7]


plot_dataset = data.frame(Y_posterior_mu,CI_2.5,CI_97.5,
                          date = as.Date(data_ohio_pm25$Date[1:366])
,
                          ORIG = data_ohio_pm25$pm2.5[1:366])


ggplot(plot_dataset)+
 geom_line(aes(x = date, y = Y_posterior_mu, colour = "predicted act
ual measurement"))+
  geom_line(aes(x = date, y = ORIG, colour = 'recorded observation')
)+
  geom_line(aes(x = date, y = CI_2.5,colour = 'CI'),linetype = "dash
ed")+
  geom_line(aes(x = date, y = CI_97.5), colour = 'red',linetype = "d
ashed")+
  labs(title = 'Posterior mean and Original data', y = 'PM 2.5 measu
rement')
```



From the plot we observe very large credible interval in regions with missing data. This means great uncertainty in predicted Y measurement which is the actual PM2.5 without measurement error. We also can observe that the recorded observation follows along the actual measure with some variance which is considered random.

To create a model which predicts the PM2.5 value for the 1st week of 1989 we need a modify the data set. This is done by adding 7 NAs after the year 1988. Then we provide initial values for these NA's the same way we did before while handling missing data. To get the RMSE between the predicted data and the original measured PM 2.5, we modify the model by adding the RMSE equation. This is then added to the list of parameters of interest to extract RMSE posterior distribution.

```
# MODEL
set.seed(123)
jags.mod.ohio_p1989 <- function(){
  # Observation model
  for (i in 2 : N) {
    Ohio[i] ~ dnorm(Y[i],tau.v)
  }
  Ohio[1] ~ dnorm(Y[1],tau.v)
  tau.v ~ dgamma(1,0.01)
  # System model
  for(i in 2:N){
    Y[i] ~ dnorm(Y[i-1],tau.w)
  }
  Y[1] ~ dnorm(6,0.001)
  tau.w ~ dgamma(1,0.01)
  sigma.w <- 1/sqrt(tau.w)
  RMSE <- sqrt((sum((Ohio[367:373]-org_ohio[367:373])^2))/7)
}


#DATA:
N = 373
org_ohio = data_ohio_pm25$pm2.5
Ohio = c(data_ohio_pm25$pm2.5[1:366],rep(NA,7))

#df_test = data.frame(withNA = Ohio,org_ohio[1:373], check = Ohio2)
data.1989 <- list("Ohio","N","org_ohio")


# initial value
inits1989_1 <- list(tau.v = 0.08, Y= runif(373,5.95,6.05),
               tau.w = 0.049,Ohio = c(rep(NA,96),rep(6.269,36),
                                      rep(NA,37),6.269,rep(NA,50),
                                      rep(6.269,5),rep(NA,53),6.269,
NA,
                                      rep(6.269,3),rep(NA,83),rep(6.
269,7)))

inits1989_2 <- list(tau.v = 0.004, Y= runif(373,5.95,6.05),
               tau.w = 0.01,Ohio = c(rep(NA,96),rep(5.6,36),
                                     rep(NA,37),5.6,rep(NA,50),
                                     rep(5.6,5),rep(NA,53),5.6,NA,
                                     rep(5.6,3),rep(NA,83),rep(5.6,7
)))

inital_val_1989 = list(inits1989_1,inits1989_2)

# param of interest
ohio_param_save = c("Ohio")

#PREDICTION:
jags.mod.p1989 <- jags(data = data.1989, inits = inital_val_1989,
                       parameters.to.save = ohio_param_save, n.chains
= 2,
                       n.iter = 10000,n.burnin = 5000,
                       n.thin=1,model.file = jags.mod.ohio_p1989 )

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 320
##    Unobserved stochastic nodes: 428
##    Total graph size: 1400
```

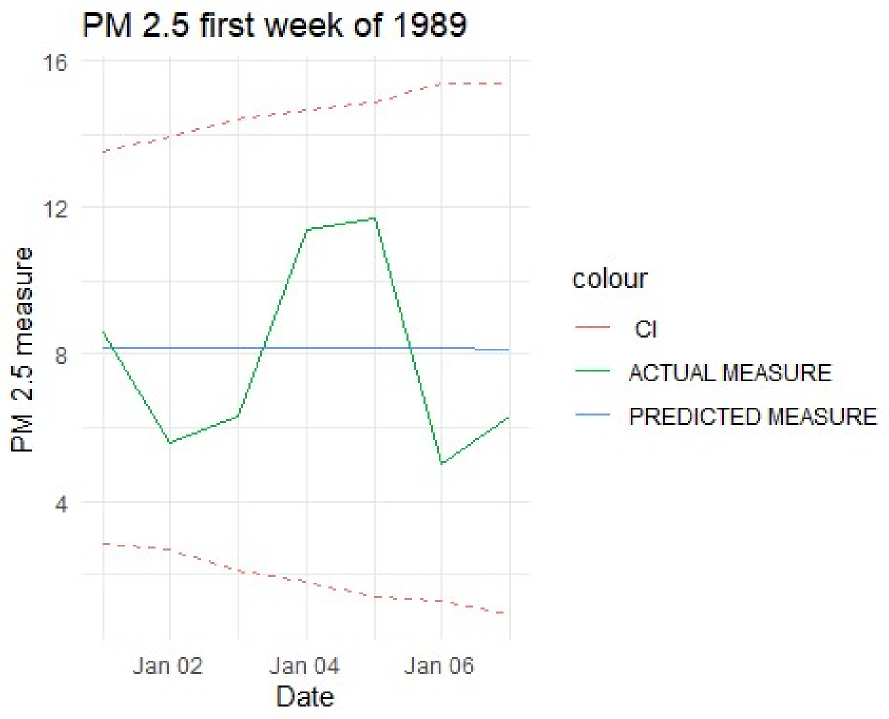
```
##
## Initializing model
```

```
jags.mod.p1989$BUGSoutput$summary[367:373,]
```

```
##                 mean       sd      2.5%      25%      50%       75%
97.5%
## Ohio[367] 8.161249 2.744285 2.8332651 6.319544 8.164389  9.997289
13.54531
## Ohio[368] 8.188312 2.921806 2.6522358 6.189110 8.142217 10.181962
13.92775
## Ohio[369] 8.155336 3.097323 2.0999951 6.071327 8.066633 10.213506
14.39765
## Ohio[370] 8.160903 3.287254 1.8055530 5.944055 8.147799 10.341940
14.64383
## Ohio[371] 8.167305 3.427221 1.3823149 5.920502 8.137228 10.419265
14.88657
## Ohio[372] 8.178616 3.602110 1.2888551 5.822525 8.175004 10.527412
15.38521
## Ohio[373] 8.143107 3.715782 0.9117309 5.711788 8.131694 10.579387
15.39063
##                 Rhat n.eff
## Ohio[367] 1.000959 10000
## Ohio[368] 1.000907 10000
## Ohio[369] 1.000933 10000
## Ohio[370] 1.001108  7100
## Ohio[371] 1.000899 10000
## Ohio[372] 1.000944 10000
## Ohio[373] 1.000924 10000
```

From the summary of the model, we get the point estimate mean of the posterior distributions and the credible intervals. We use this to plot a graph of the original measured and predicted PM 2.5.

```
# data set for plotting
pred_ohio_dataset = data.frame(pred1 = jags.mod.p1989$BUGSoutput$sum
mary[367:373,1],
                               CI_2.5_p = jags.mod.p1989$BUGSoutput$
summary[367:373,3],
                               CI_97.5_p = jags.mod.p1989$BUGSoutput
$summary[367:373,7],
                               org = data_ohio_pm25[367:373,]$pm2.5,
                               date = as.Date(data_ohio_pm25[367:373
,]$Date))

#PLOT
ggplot(pred_ohio_dataset)+
  geom_line(aes(x = date, y = pred1, colour = 'PREDICTED MEASURE'))+
  geom_line(aes(x = date, y = org, colour = 'ACTUAL MEASURE'))+
  geom_line(aes(x = date, y = CI_2.5_p, colour = ' CI'),linetype = "
dashed")+
  geom_line(aes(x = date, y = CI_97.5_p), colour = ' indianred1',lin
etype = "dashed")+
  labs(title = "PM 2.5 first week of 1989", y = 'PM 2.5 measure', x
= 'Date')+
 theme_minimal()
```



Even though the actual measured value deviates from the predicted value the plot shows that the prediction along with the credible interval captures the original measured PM 2.5. Therefore our model predicts well.

```
#RMSE :
ohio_param.RMSE = c('RMSE')


jags.mod.p1989.RMSE <- jags(data = data.1989, inits = inital_val_198
9,
                     parameters.to.save = ohio_param.RMSE, n.chains
= 2,
                     n.iter = 10000,n.burnin = 5000,
                     n.thin=1,model.file = jags.mod.ohio_p1989 )

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 320
##    Unobserved stochastic nodes: 428
##    Total graph size: 1400
```

```
##
## Initializing model
```

```
print(jags.mod.p1989.RMSE)
```

```
## Inference for Bugs model at "C:/Users/JUANJO~1/AppData/Local/Temp
/RtmpgvcmIu/model7824335c5067.txt", fit using jags,
##  2 chains, each with 10000 iterations (first 5000 discarded)
##  n.sims = 10000 iterations saved
##           mu.vect sd.vect     2.5%      25%      50%      75%
97.5%  Rhat
## RMSE        4.035   1.155    2.215    3.260    3.894    4.638
6.742 1.001
## deviance 1399.709  70.825 1250.171 1353.711 1405.773 1452.505 151
1.966 1.007
##          n.eff
## RMSE     10000
## deviance   390
##
## For each parameter, n.eff is a crude measure of effective sample
size,
## and Rhat is the potential scale reduction factor (at convergence,
Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 2502.3 and DIC = 3902.0
## DIC is an estimate of expected predictive error (lower deviance i
s better).
```

From the posterior distribution of RMSE, we get the estimated Root mean square error between the original and predicted PM 2.5. RMSE is used to measure the difference between the predicted and observed values. This is a measure of the accuracy of the model. Here get an RMSE of 4.035 this is the residual for the prediction. This is high for this dataset but the credible interval captures the original data.

When we have data of measurements and as we already produced posterior without the measurement, the priors do not have to be vague. The priors can come from an informative prior with a low standard deviation. Therefore the precision parameters should be high. In this model, we can have tau. v and tau.w, which is the precision parameter for Ohio, come from dgamma(1, 0.001) this will be an exponential distribution with most of the values centred around 1000. As we repeated the analysis with new data we can use the posterior of the previous model without the measurements a prior for the new model.