# Bayes Data Analysis Assignment 02

**Problem 01**

**01-(a)**

Exploratory analysis: Define a ratio = deaths/rep.events has a decreasing trend over the periods considered. During 1986-1993 and 1994-2003, when reported events increased suddenly, ratio would rush up, even exceed over 1 in 1991 and 1994, therefore this ratio fluctuated a lot during 1986-1993 and 1994-2003. But during 2004-2019, the ratio decreases to a low level (around 0.4) but still has little bumps.

1991 has the largest deaths and corresponding ratio, 2005 has the largest reported events but has the lowest ratio.

Ratio line plot is given in Figure 1. Line plot of reported events and deaths is given in Figure 2.

手机屏幕截图

描述已自动生成

Table 1: dataset rows relative to the years 1986, 1994, 2004.

许多照片放在一起

描述已自动生成

Figure 1: Line plot of death ratio (deaths/reported events)

图片包含 游戏机

描述已自动生成

Figure 2: Temporal evolution of the number of avalanches and deaths

**01-(b)**

Parameter interpretation: Consider the analysis of the relationship between the number of Deaths (response variable) and the covariates “Reported events” and two dummies “EADS1”, “EADS2”. I set the following generalized linear model with a logarithm link connecting the response mean with the linear combination of the covariates:

(1)

where represents the centered reported events. The second line of Equation (1) highlights the multiplicative effect of the covariates on the mean .

After running JAGS, Brooks-Gelman-Rubin statistic of are all around 1 and there’s no trend in trace plots, no severe lag in ACF plots, which indicates MCMC converged.

手机屏幕截图

描述已自动生成 图片包含 游戏机, 截图

描述已自动生成

Figure 3: Trace plots and Gelman plots for parameters in model 01-(b)

Interpretations of results are as follow:

* =3.45 is the expected number of deaths for a year between 1986 -1993.
* A year with a reported events one unit more than the mean has an expected number of deaths increased of 1.22, while the other variables are hold constant.
* A year between 1994-2003 has an expected number of deaths increased of 0.91, while the other variables are hold constant.
* A year after 2004 has an expected number of deaths increased of 0.41, while the other variables are hold constant.

**01-(c)**

1. The probability of observing <15 when reported events = 20, year = 2020 (thus EADS1=0, EADS2 = 1) is 0.188.
2. Posterior mean of >1 in period1 (EADS1=0, EADS2=0) is 0.82; in period2 (EADS1=1, EADS2 = 0) is 0.67; in period3 (EADS1=0, EADS2 =1) is 0.018. Probabilities shrink through these periods.

**01-(d)**

Calculation:

* Based on , therefore:

(2)

Where is Reported events,,,,.

(2) can be simplified into:

that is to say .

* Similarly, I derive pdf of :

That is to say .

* According to experts, is usually between 5 and 15, I assume this means expectation of is 10 and standard deviation is 5; is between and 4, I assume this means expectation of is 2.1 and standard deviation is 1.87. My computation base on indicates that expectation of is ，which means if lognormal prior for is appropriate, expert’s expectation of (which is 10) will nearly equal to , then , it’s not consistent with experts opinion of , besides, standard deviation is not consistent with standard deviation (which is 5) from experts, either.

Similarly, if the given lognormal prior for is sensitive, then I derived should be around 3.87, this value is not consistent with expert’s opinion of . Therefore isn’t an appropriate prior.

Simulation: Besides, I carried out a simulation by rjags, converged simulation results are as follow:

* The probability of value of less than -4 or greater than 4: 0.52
* The probability of absolute value of less than : 0.39

Simulation results are consistent with computations, indicates that isn’t an appropriate prior.

**01-(e)**

Expand the previous model in (1b) by including an extra variance term . I set the following generalized linear model with a logarithm link connecting the response mean with the linear combination of the covariates and :

(2)

where represent the high variability in the number of deaths for different years.

After running JAGS, similar with convergence diagnosis in 01-(b), Brooks-Gelman-Rubin statistic of and are all around 1 and there’s no trend in their trace plots, which indicates MCMC converged. Comparisons are as follow:

* =2.58 is the expected number of deaths for a year between 1986 and 1993. =3.45 is the expected number of deaths for a year between 1986 and 1993.
* A year with a reported events one unit more than the mean has an expected number of deaths increased of =1.26, 1.22, while the other variables are hold constant.
* A year between 1994-2003 has an expected number of deaths increased of 1.04,= 0.91, while the other variables are hold constant.
* A year after 2004 has an expected number of deaths increased of = 0.53, 0.41, while the other variables are hold constant.

**01-(f)**

图片包含 游戏机

描述已自动生成

Figure 4: posterior means with 90% point-wise credible intervals for model in (b)

图片包含 游戏机, 房间

描述已自动生成

Figure 5: posterior means with 90% point-wise credible intervals for model in (e)

Deviance Information Criterion of model in (b) is 137.9, model in (e) is 110, difference between them is 13.34, Therefore, based on DIC the model in (e) is better.

Plots in (a) and (f) indicates that model in (e) captures more fluctuation and predicted death numbers are more accurate. Death number in Figure 4 seems more similar to Figure 2 (Actual Deaths), especially during 2004-2019. For example, in 2005, Figure 3 shows that deaths rush up, but Figure 4 doesn’t, which is more consistent with the actual death number. However, 90% credit interval band of Figure 4 is wider than Figure 3, this is because the introduction of in model (e).

Sum up what are discussed afore, model in (e) is better from my point of view.

**Problem 02**

**02-(a)**

手机屏幕截图

描述已自动生成

Table 2: Avalanches part 2 data frame head

图片包含 游戏机

描述已自动生成

Figure 6: Boxplot for Snow\_days and Snow\_total

手机屏幕截图

描述已自动生成

Table 3: Correlation matrix of Season, Snow\_total and Snow\_days

Correlation between Snow\_days and Snow\_total is 0.8, which indicates there’s collinearity if they are both in a general linearly combined model.

**02-(b)**

图片包含 游戏机, 钟表

描述已自动生成

Figure 7: DAG of Hierarchical Model in 2-(b)

Hierarchical formulas are as follow:

Where , and are centered covariates, is the category of Geo\_space corresponding random effect, is the number of hits, is the probability of deaths. After running rJAGS, Brooks-Gelman-Rubin statistic of and are all around 1 and there’s no trend in trace plots, which indicates MCMCs are converged. But and ACF plots are lagged. Results interpretations are as follow:

An event with year one unit more than the mean has an expected log odds of death probability changed of , while the other variables are hold constant.

An event with Snow\_total one unit more than the mean has an expected log odds of death probability changed of , while the other variables are hold constant.

An event with year one unit more than the mean has an expected log odds of death probability changed of , while the other variables are hold constant.

,, are expected log odds of death probability with random effect (corresponding to three different Geo\_space) when Season, Snow\_total, Snow\_day are equal to their mean.

**02-(c)**

手机屏幕截图

描述已自动生成 社交网站的手机截图

描述已自动生成

Figure 8: Trace plots and Gelman Plots in Hierarchical model in 02-(b)

手机屏幕截图

描述已自动生成 手机截图图社交软件的信息

描述已自动生成

Figure 9: Trace plots and Gelman Plots in Hierarchical model in 02-(b)

Trace plots of and are not as completely random as trace plot of in Hierarchical model in 02-(b), and their corresponding ACF plots are obviously lagged. But trace plots of slopes in model in 02-(c) are random and ACF is not lagged, which means MCMCs in model in 02-(c) converge better. This is because Snow\_Total and Snow\_Days are highly correlated. Model in 02-(c) fixed this collinearity problem by delete Snow\_Days. Results interpretations from model in 02-(c) are as follow:

An event with year one unit more than the mean has an expected log odds of death probability changed of , while the other variables are hold constant.

An event with Snow\_total one unit more than the mean has an expected log odds of death probability changed of , while the other variables are hold constant.

and aren’t changed a lot, this is also because Snow\_Days and Snow\_Total are highly correlated, Snow\_Day doesn’t provide new “information” to model.

,,, they are similar to results in 02-(c).

**02-(d)**

Based on model obtained in (c), I estimated the posterior expected value and 95% credible interval of the proportion of deaths near recording stations 1, 8 and 10 for the Seasons 2015 and 2018. Specific values are given in Table 4.

手机屏幕截图

描述已自动生成

Table 4: posterior expected value and 95% credit interval 2(d)

According to Table 4, proportion of death decrease as year increase (while other covariates remain the same). Besides, proportion of death in space 3 is lower than space 1 and 2; space1 and 2 have similar proportion of death (while other covariates remain the same).

Compare the probability of a proportion of deaths greater than 60%: station 10< station 1 < station 8, Year 2018<Year 2015.

**02-(e)**

Set up a new hierarchical model where the random effects are placed on the recording stations (Rec.station), after diagnosis, all parameters and random effects converge well. DIC of model in 2(c) is 87.9, model in 2(d) is 75.36, based on DIC, model in 2(d) performs better.

手机屏幕截图

描述已自动生成

Table 5: posterior expected value and 95% credit interval 2(e)

Proportions of death (posterior expected value) increase because model in 2(d) provides a better fit. And lengths of credit intervals are longer compared with credit intervals in Table 4, this is probably because place random effects on a more specific category variable increase variability of model,

**02-(f)**

In order to capture variability in the different recording stations and mountain areas in one single hierarchical model, we can employ another random effect based on model in 2(e).

电脑游戏的截图

描述已自动生成

Figure 10: DAG of Hierarchical Model in 2-(f)

Hierarchical formulas are as follow:

Where is the category of Geo\_space corresponding random effect, is the category of Rec.station corresponding random effect, k = 1,2…,10,11. Each has two random effects, one from , the other from . Other symbols meanings are similar to DAG in 2(b)

**R code：**

---

title: "BayesA02"

author: "Wenyi Fang"

date: "4/6/2020"

output:

word\_document: default

html\_document: default

---

#read data

```{r}

setwd("/Users/vanessafung/Desktop")

Avalanches = read.csv("Avalanches.csv",header = TRUE, sep=";",fileEncoding="UTF-8-BOM")

Avalanches\_part2<-read.csv("Avalanches\_part2.csv",header = TRUE, sep=";",fileEncoding="UTF-8-BOM")

```

#libray packages

```{r}

library(MCMCpack)

library(magrittr)

require(rjags)

library(coda)

library(tidyverse)

library(tibble)

library(dplyr)

library(tidyr)

library(purrr)

library(ggplot2)

library(data.table)

library(caret)

library(lmtest)

library(runjags)

library(ggcorrplot)

```

#Problem 01

#01-(a)Add to the dataset two dummy variables called EADS1 and EADS2, exploratory analysis

#any other way of exploratory?

#comment: ratio = deaths/rep.events has a decreasing trend,during 1986-1993 and 1994-2003,when reported events increased suddenly, deaths would rush up,even exceed over 1 in 1991 and 1994, therefore death rate fluctuated a lot during 1986-1993 and 1994-2003. But during 2004-2019, death rate decrease to a low level （around 0.4) and fluctuate less

```{r}

#define dummies

Avalanches[, "EADS1"] = if\_else(Avalanches$Season <= 2003 &

Avalanches$Season >= 1994, 1, 0)

Avalanches[, "EADS2"] = if\_else(Avalanches$Season >= 2004, 1, 0)

Avalanches[,"Season.period"] = as.character()

Avalanches[,"Season.period"][1:8] = "1986-1993"

Avalanches[,"Season.period"][9:18] = "1994-2003"

Avalanches[,"Season.period"][19:34] = "2004-"

#row relative to 1986,1994,2004

sub.3.years = Avalanches[c(

which(Avalanches$Season == 1986),

which(Avalanches$Season == 1994),

which(Avalanches$Season == 2004)

), ]

sub.3.years

#exploratory analysis

sum.avalanches = summary(Avalanches[, c(2, 3)])

sum.avalanches

corr.event.death = cor(Avalanches[, 2], Avalanches[, 3])

corr.event.death

death.ratio.trend = ggplot(Avalanches, aes(x = Season, y = Deaths / Rep.events)) +

geom\_line() +

geom\_point() +

xlim(1986, 2019) +

geom\_vline(xintercept = c(1993, 2004),

color = "red",

alpha = 0.5)+

geom\_text(aes(x = 1993,y = 1.35,label = "1993"),col = "red")+

geom\_text(aes(x = 2004,y = 1.35,label = "2004"),col = "red")

box.periods = ggplot(Avalanches,aes(x = Season.period,y = Deaths,group = Season.period))+

geom\_boxplot( )

# geom\_boxplot(aes(x = 9:18,y = Deaths[9:18]))+

# geom\_boxplot(aes(x = 19:34,y = Deaths[19:34]))

box.periods

#graph

events.season.line = ggplot(Avalanches, aes(x = Season)) +

geom\_line(aes(y = Rep.events), color = "#FF9999") +

geom\_text(

data = Avalanches[Avalanches$Season == 2014, ],

aes(y = Rep.events, label = "Rep.events"),

color = "#CC79A7",

size = 5

) +

geom\_line(aes(y = Deaths), color = "#0072B2", alpha = 0.7) +

geom\_text(

data = Avalanches[Avalanches$Season == 2014, ],

aes(y = Deaths, label = "Deaths"),

color = "#0072B2",

size = 5

) +

geom\_area(aes(y = Deaths), fill = "#0072B2", alpha = 0.5) +

geom\_area(aes(y = Rep.events), fill = "#FF9999", alpha = 0.1) +

geom\_vline(xintercept = c(1993, 2004), color = "#D55E00") +

geom\_text(aes(x = 1993,y = 15,label = "1993"),col = "#D55E00")+

geom\_text(aes(x = 2004,y = 15),col = "#D55E00",label = "2004")+

ylab("Reported events or Deaths")

death.ratio.trend

events.season.line

```

death.ratio.trend

events.season.line

#01-(b)

```{r}

#Data block and Initial Values

n = length(Avalanches$Season)

Avalanches.data = list(

n = n,

Deaths = Avalanches$Deaths,

Events = Avalanches$Rep.events,

ind.period1 = Avalanches$EADS1,

ind.period2 = Avalanches$EADS2

)

Avalanches.inits <- list(

list(

beta0 = -1,

beta1 = -1,

beta2 = -1,

beta3 = -1

),

list(

beta0 = 0,

beta1 = 1,

beta2 = 2,

beta3 = 3

),

list(

beta0 = 3,

beta1 = 2,

beta2 = 1,

beta3 = 0

)

)

#model statement

Avalnches.model <- "model {

# Hyperparameters

beta.mu.0 <- 0

beta.tau.0 <- 0.001

# prior

beta0 ~ dnorm(beta.mu.0,beta.tau.0)

beta1 ~ dnorm(beta.mu.0,beta.tau.0)

beta2 ~ dnorm(beta.mu.0,beta.tau.0)

beta3 ~ dnorm(beta.mu.0,beta.tau.0)

#Likelihood

for(i in 1:n) {

# Note: link function on LHS of fn assignment

log(mu[i]) <- beta0+beta1\*(Events[i]-mean(Events[]))+beta2\*ind.period1[i]+beta3\*ind.period2[i]

Deaths[i] ~ dpois(mu[i])

}

}"

# Run JAGS to the completion of the "adaption" stage

results.Avalanches.A <-

jags.model(

file = textConnection(Avalnches.model),

data = Avalanches.data,

inits = Avalanches.inits,

n.chains = 3

)

# Burn-in of 10000 iterations

update(results.Avalanches.A, n.iter = 5000)

# Longer run for making inferences, assuming chains have converged

results.Avalanches.B <- coda.samples(

results.Avalanches.A,

variable.names = c("beta0", "beta1", "beta2", "beta3"),

n.iter = 10000

)

#MCMC Convergence Diagnostics

# Trace plots and density

plot(results.Avalanches.B)

# Brooks-Gelman-Rubin statistic (want a value near 1)

gelman.plot(results.Avalanches.B)

#"mixing"- effective sample size given 10,000

effectiveSize(results.Avalanches.B[[1]][, "beta0"])

effectiveSize(results.Avalanches.B[[1]][, "beta1"])

effectiveSize(results.Avalanches.B[[1]][, "beta2"])

effectiveSize(results.Avalanches.B[[1]][, "beta3"])

#ACF plot

{

autocorr.plot(results.Avalanches.B[[1]][, "beta0"], main = "Intercept")

autocorr.plot(results.Avalanches.B[[1]][, "beta1"], main = "Slope1")

autocorr.plot(results.Avalanches.B[[1]][, "beta2"], main = "Slope2")

autocorr.plot(results.Avalanches.B[[1]][, "beta3"], main = "Slope3")

}

#look at MCMC summary

summary(results.Avalanches.B)

fit.events.b = as.data.frame(combine.mcmc(results.Avalanches.B))

mean(exp(fit.events.b$beta0))# 3.419887

mean(exp(fit.events.b$beta1))#1.216833

mean(exp(fit.events.b$beta2))#0.9231521

mean(exp(fit.events.b$beta3))# 0.4125329

```

#01-(c)

```{r}

#post\_mu when rep.events = 20, EADS1=0,EADS2 = 1

post\_mu = exp(

fit.events.b$beta0 + fit.events.b$beta1 \* (20 - mean(Avalanches$Rep.events)) +

fit.events.b$beta2 \* 0 + fit.events.b$beta3 \* 1

)

mean(post\_mu)

quantile (post\_mu,c(0.025,0.975))

#prob of mu<15

prob.less.15 = length(which(post\_mu < 16)) / length(post\_mu)

prob.less.15

#c-2-Solution1

#set event = 1,a constant number

#0,0

post\_mu2 = exp(

fit.events.b$beta0 + fit.events.b$beta1 \* (1 - mean(Avalanches$Rep.events)) + fit.events.b$beta2 \*0 + fit.events.b$beta3 \* 0

)

prob.more.1.period1 = length(which(post\_mu2 > 1)) / length(post\_mu2)

prob.more.1.period1#0.82

#1,0

post\_mu3 = exp(

fit.events.b$beta0 + fit.events.b$beta1 \* (1 - mean(Avalanches$Rep.events)) + fit.events.b$beta2 \*

1 + fit.events.b$beta3 \* 0

)

prob.more.1.period2 = length(which(post\_mu3 > 1)) / length(post\_mu3)

prob.more.1.period2#0.67

#0,1

post\_mu4 = exp(fit.events.b$beta0 + fit.events.b$beta1 \* (1 - mean(Avalanches$Rep.events)) + fit.events.b$beta2 \*

0 + fit.events.b$beta3 \* 1)

prob.more.1.period3 = length(which(post\_mu4 > 1)) / length(post\_mu4)

prob.more.1.period3#0.018

#c-2-solution2

mu.eachevent = function(ind1,ind2){

logmu = as.numeric()

for (i in 1:length(Avalanches$Rep.events)){

logmu[i] = fit.events.b$beta0 + fit.events.b$beta1 \* (Avalanches$Rep.events[i]-mean(Avalanches$Rep.events)) + fit.events.b$beta2 \*ind1 + fit.events.b$beta3 \* ind2

}

return(exp(logmu))

}

#calculate each obs average deaths (mu) based on beta estimation

mu.period1 = mu.eachevent(0,0)

mu.period2 = mu.eachevent(1,0)

mu.period3 = mu.eachevent(0,1)

#calculate group mean of mu for each periods as the estimate of poisson dist parameter

mu.period1.mean = mean(mu.period1)

mu.period2.mean = mean(mu.period2)

mu.period3.mean = mean(mu.period3)

#use qpois to calculate probability

prob.more.1.period1 = 1-ppois(1,mu.period1.mean)

prob.more.1.period2 = 1-ppois(1,mu.period2.mean)

prob.more.1.period3 = 1-ppois(1,mu.period3.mean)

prob.more.1.period1;prob.more.1.period2;prob.more.1.period3

```

#01-(d)

```{r}

# ---- Bayesian model

# Poisson Model

Avalanches.d.data <- list(n = nrow(Avalanches), Deaths = Avalanches$Deaths, EADS1 = Avalanches$EADS1, EADS2 = Avalanches$EADS2)

# Create initial values for JAGS

num.chains <- 3

Avalanches.d.inits <- function(){

log\_fi <- rnorm(1,0,10)

beta0 <- rnorm(1,0,10)

beta1 <- rnorm(1,0,10)

beta2 <- rnorm(1,0,10)

return( list(log\_fi = log\_fi, beta0=beta0, beta1=beta1, beta2=beta2) )

}

# Create model block for JAGS

Avalanches.d.model <- "model {

# prior

log\_fi ~ dnorm(0, 0.25)

beta0 ~ dnorm(0, 0.001)

beta1 ~ dnorm(0, 0.001)

beta2 ~ dnorm(0, 0.001)

#Likelihood

for(i in 1:n) {

log(mu[i]) <- beta0 + log\_fi + beta1\*EADS1[i] + beta2\*EADS2[i]

Deaths[i] ~ dpois(mu[i]) }

}"

# Run JAGS to the completion of the "adaption" stage

results.Avalanches.d.A <- jags.model(file = textConnection(Avalanches.d.model), data = Avalanches.d.data, inits = Avalanches.d.inits, n.chains = num.chains, quiet = TRUE)

update(results.Avalanches.d.A, n.iter = 20000)

results.Avalanches.d.B <- coda.samples(results.Avalanches.d.A, variable.names = "log\_fi", n.iter = 50000)

# posterior estimates for log\_phi

fit.Avalanches.d<- as.data.frame(combine.mcmc(results.Avalanches.d.B))

cat("The probability of absolute value of log\_phi less than 4: ", mean(abs(fit.Avalanches.d$log\_fi) <= log(4)), "\n")

length(which(fit.Avalanches.d$log\_fi>=5&fit.Avalanches.d$log\_fi<=15))/length(fit.Avalanches.d$log\_fi)

# posterior estimates for beta\_rep.events

beta\_posterior <- as.matrix(1/(Avalanches$Rep.events-mean(Avalanches$Rep.events)))%\*%t(as.matrix(fit.Avalanches.d$log\_fi))

cat("The probability of absolute value of beta\_repevents less than log(4)/5: ", mean(abs(beta\_posterior) <= log(4)/5))

exp(-4)

```

#01-(e)

```{r}

#Data block and Initial Values

n = length(Avalanches$Season)

Avalanches.data = list(n=n,Deaths = Avalanches$Deaths, Events = Avalanches$Rep.events,ind.period1 = Avalanches$EADS1,ind.period2 = Avalanches$EADS2)

Avalanches.extratheta.inits = list(

list(beta0 = -1,

beta1 = -1,

beta2 = -1,

beta3 = -1),

list(beta0 = 0,

beta1 = 1,

beta2 = 2,

beta3 = 3),

list( beta0 = 3,

beta1 = 2,

beta2 = 1,

beta3 = 0)

)

#Extra variance model statement

Avalnches.extra.variance.model <- "model {

# Hyperparameters

beta.mu.0 <- 0

beta.tau.0 <- 0.01

theta.mu.0 <- 0

theta.se ~ dunif(0,10)

theta.tau <- 1/pow(theta.se,2)

# prior

beta0 ~ dnorm(beta.mu.0,beta.tau.0)

beta1 ~ dnorm(beta.mu.0,beta.tau.0)

beta2 ~ dnorm(beta.mu.0,beta.tau.0)

beta3 ~ dnorm(beta.mu.0,beta.tau.0)

#Likelihood

for(i in 1:n) {

log(mu[i]) <- beta0+beta1\*(Events[i]-mean(Events[]))+beta2\*ind.period1[i]+beta3\*ind.period2[i]+theta[i]

Deaths[i] ~ dpois(mu[i])

theta[i] ~ dnorm(theta.mu.0,theta.tau)

}

}"

# Run JAGS to the completion of the "adaption" stage

results.Avalanches.extratheta.A <-

jags.model(

file = textConnection(Avalnches.extra.variance.model),

data = Avalanches.data,

inits = Avalanches.extratheta.inits,

n.chains = 3

)

# Burn-in of 10000 iterations

update(results.Avalanches.extratheta.A, n.iter = 10000)

# Longer run for making inferences, assuming chains have converged

results.Avalanches.extratheta.B <- coda.samples(

results.Avalanches.extratheta.A,

variable.names = c("beta0", "beta1", "beta2", "beta3","theta"),

thin = 50,

n.iter = 50000

)

#MCMC Convergence Diagnostics

# Trace plots and density

plot(results.Avalanches.extratheta.B)

# Brooks-Gelman-Rubin statistic (want a value near 1)

gelman.plot(results.Avalanches.extratheta.B)

#"mixing"- effective sample size given 10,000

effectiveSize(results.Avalanches.extratheta.B[[1]][, "beta0"])

effectiveSize(results.Avalanches.extratheta.B[[1]][, "beta1"])

effectiveSize(results.Avalanches.extratheta.B[[1]][, "beta2"])

effectiveSize(results.Avalanches.extratheta.B[[1]][, "beta3"])

#ACF plot

{autocorr.plot(results.Avalanches.extratheta.B[[1]][, "beta0"], main = "Intercept")

autocorr.plot(results.Avalanches.extratheta.B[[1]][, "beta1"], main = "Slope1")

autocorr.plot(results.Avalanches.extratheta.B[[1]][, "beta2"], main = "Slope2")

autocorr.plot(results.Avalanches.extratheta.B[[1]][, "beta3"], main = "Slope3")

}

#look at MCMC summary

summary(results.Avalanches.extratheta.B)

fit.events.e = as.data.frame(combine.mcmc(results.Avalanches.extratheta.B))

mean(exp(fit.events.e$beta0))# 3.419887

mean(exp(fit.events.e$beta1))#1.216833

mean(exp(fit.events.e$beta2))#0.9231521

mean(exp(fit.events.e$beta3))# 0.4125329

```

#01-(f)

```{r}

#1(b)

#calculate average of death for each row

mean.repevents = mean(Avalanches$Rep.events)

post.mu.b = exp(as.matrix(fit.events.b)%\*%t(as.matrix(cbind(rep(1,34),Avalanches[,2]-mean.repevents,Avalanches[,c(4,5)]))))

#1(e)

#calculate average of death for each row

post.mu.e = exp(as.matrix(fit.events.e[,1:4])%\*%t(as.matrix(cbind(rep(1,34),Avalanches[,2]-mean.repevents,Avalanches[,c(4,5)])))+as.matrix(fit.events.e[,5:38]))

#plots

post.mu.mean.b = colMeans(post.mu.b)

post.mu.mean.e = colMeans(post.mu.e)

quantile.90 = function(x){

quantile(x,c(0.05,0.95))

}

post.mu.quantile.b = apply(post.mu.b, 2, quantile.90)

post.mu.quantile.e = apply(post.mu.e, 2, quantile.90)

#plot for b

post.avalanches.b.df = data.frame(

Avalanches,

post.mu.mean.b,

quantile.05 = post.mu.quantile.b[1,],

quantile.95 = post.mu.quantile.b[2,]

)

post.mu.graph.b = ggplot(post.avalanches.b.df,aes(x=Season,y = post.mu.mean.b))+

geom\_line(col = "#0072B2") +

geom\_line(aes(y = quantile.05),linetype = "longdash",col = "#D55E00")+

geom\_line(aes(y = quantile.95),linetype = "longdash",col = "#D55E00")

post.mu.graph.b

#plot for e

post.avalanches.e.df = data.frame(

Avalanches,

post.mu.mean.e,

quantile.05 = post.mu.quantile.e[1,],

quantile.95 = post.mu.quantile.e[2,]

)

post.mu.graph.e = ggplot(post.avalanches.e.df,aes(x=Season,y = post.mu.mean.e))+

geom\_line(col = "#0072B2") +

geom\_line(aes(y = quantile.05),linetype = "longdash",col = "#D55E00")+

geom\_line(aes(y = quantile.95),linetype = "longdash",col = "#D55E00")

post.mu.graph.e

#DIC p18

dic.no.theta <- dic.samples(model=results.Avalanches.A,n.iter=10000,type="pD")

dic.w.theta <- dic.samples(model=results.Avalanches.extratheta.A,n.iter=10000,type="pD")

diffdic(dic.no.theta,dic.w.theta)#positive ,the second one is better

events.season.line

post.mu.graph.b

post.mu.graph.e

```

#Problem02

#02-(a)

```{r}

Avalanches\_part2$Snow\_total = Avalanches\_part2$Snow\_total/100

Avalanches\_part2$Snow\_days = Avalanches\_part2$Snow\_days/14

head(Avalanches\_part2,5)

#plot

a1 = ggplot(Avalanches\_part2,aes(x = Season,y=death.rate,group = Season))+

geom\_boxplot(aes(y = Snow\_total),col = "#0072B2")+

geom\_boxplot(aes(y = Snow\_days),col = "#FF9999")+

ylab("Snow\_total or Snow\_days")+

geom\_text(

data = Avalanches\_part2[Avalanches\_part2$Season == 2019, ],

aes(y = 14, label = "Snow Days"),

color = "#FF9999",

size = 5

)+

geom\_text(

data = Avalanches\_part2[Avalanches\_part2$Season == 2019, ],

aes(y = 13, label = "Snow Total"),

color = "#0072B2",

size = 5

)

a1

corr <- round(cor(Avalanches\_part2[,c(2,7,8)]), 1)

a2 = ggcorrplot(corr)

a2

```

#2(b)

```{r}

# Data block #includes hyperparameters

n = length(Avalanches\_part2$Event\_ID)

death.hier.b.data <- list(n=n,Season=Avalanches\_part2$Season,

Snow\_total=Avalanches\_part2$Snow\_total,Snow\_days = Avalanches\_part2$Snow\_days,hit = Avalanches\_part2$Hit,

Geo.space=Avalanches\_part2$Geo\_space,y.dead = Avalanches\_part2$Deaths,

J=max(Avalanches\_part2$Geo\_space))

# Initial values

death.hier.b.inits <- function(){

list(fi=rnorm(max(Avalanches\_part2$Geo\_space), 0,runif(1,0,10)),

beta1=rnorm(1,0,sqrt(10)),

beta2=rnorm(1,0,sqrt(10)),

beta3=rnorm(1,0,sqrt(10)))

}

deathrate.b.model <- "model{

#likelihood

for(i in 1:n) {

logit(mu[i]) <- fi[Geo.space[i]] + beta1\*(Season[i] - mean(Season[])) + beta2\*(Snow\_total[i] - mean(Snow\_total[])) + beta3\*(Snow\_days[i]- mean(Snow\_days[]))

y.dead[i] ~ dbin(mu[i],hit[i])

}

#Prior for random effect

for(j in 1:J){

fi[j] ~ dnorm(0,1/pow(fi.se,2))

}

#hyperparameters and hyperprior

beta.mu.0= 0

beta.tau.0 = 0.1

fi.se ~ dunif(0,10)

#prior for beta

beta1 ~ dnorm(beta.mu.0,beta.tau.0)

beta2 ~ dnorm(beta.mu.0,beta.tau.0)

beta3 ~ dnorm(beta.mu.0,beta.tau.0)

}"

death.hier.res.b.A <- jags.model(file=textConnection(deathrate.b.model), data=death.hier.b.data, inits=death.hier.b.inits, n.chains=3, quiet = TRUE)

update(death.hier.res.b.A, n.iter=2000)

death.hier.res.b.B <- coda.samples(death.hier.res.b.A,

variable.names=c("fi","beta1","beta2","beta3"), n.iter=10000)

```

```{r}

#MCMC Convergence Diagnostics

# Trace plots and density

plot(death.hier.res.b.B)

# Brooks-Gelman-Rubin statistic (want a value near 1)

gelman.plot(death.hier.res.b.B)

#"mixing"- effective sample size given 10,000

effectiveSize(death.hier.res.b.B[[1]][, "beta1"])

effectiveSize(death.hier.res.b.B[[1]][, "beta2"])

effectiveSize(death.hier.res.b.B[[1]][, "beta3"])

effectiveSize(death.hier.res.b.B[[1]][, "fi[1]"])

effectiveSize(death.hier.res.b.B[[1]][, "fi[2]"])

effectiveSize(death.hier.res.b.B[[1]][, "fi[3]"])

#ACF plot

{

autocorr.plot(death.hier.res.b.B[[1]][, "beta1"], main = "Slope1")

autocorr.plot(death.hier.res.b.B[[1]][, "beta2"], main = "Slope2")

autocorr.plot(death.hier.res.b.B[[1]][, "beta3"], main = "Slope3")

autocorr.plot(death.hier.res.b.B[[1]][,"fi[1]"], main = "random effect 1")

autocorr.plot(death.hier.res.b.B[[1]][,"fi[2]"], main = "random effect 2")

autocorr.plot(death.hier.res.b.B[[1]][,"fi[3]"], main = "random effect 3")

}

#look at MCMC summary

summary(death.hier.res.b.B)

death.hier.b = as.data.frame(combine.mcmc(death.hier.res.b.B))

```

#02-(c)

```{r}

# Data block #includes hyperparameters

n = length(Avalanches\_part2$Event\_ID)

death.hier.c.data <- list(n=n,Season=Avalanches\_part2$Season,

Snow\_total=Avalanches\_part2$Snow\_total,hit = Avalanches\_part2$Hit,

Geo.space=Avalanches\_part2$Geo\_space,y.dead = Avalanches\_part2$Deaths,

J=max(Avalanches\_part2$Geo\_space))

# Initial values

death.hier.c.inits <- function(){

list(fi=rnorm(max(Avalanches\_part2$Geo\_space), 0,runif(1,0,10)),

beta1=rnorm(1,0,sqrt(10)),

beta2=rnorm(1,0,sqrt(10)))

}

deathrate.c.model <- "model{

#likelihood

for(i in 1:n) {

logit(mu[i]) <- fi[Geo.space[i]] + beta1\*(Season[i] - mean(Season[])) + beta2\*(Snow\_total[i] - mean(Snow\_total[]))

y.dead[i] ~ dbin(mu[i],hit[i])

}

#Prior for random effect

for(j in 1:J){

fi[j] ~ dnorm(0,1/pow(fi.se,2))

}

#hyperparameters and hyperprior

beta.mu.0= 0

beta.tau.0 = 0.1

fi.se ~ dunif(0,10)

#prior for beta

beta1 ~ dnorm(beta.mu.0,beta.tau.0)

beta2 ~ dnorm(beta.mu.0,beta.tau.0)

}"

death.hier.res.c.A <- jags.model(file=textConnection(deathrate.c.model), data=death.hier.c.data, inits=death.hier.c.inits, n.chains=3, quiet = TRUE)

update(death.hier.res.c.A, n.iter=2000)

death.hier.res.c.B <- coda.samples(death.hier.res.c.A,

variable.names=c("fi","beta1","beta2"), n.iter=10000)

```

```{r}

#MCMC Convergence Diagnostics

# Trace plots and density

plot(death.hier.res.c.B)

# Brooks-Gelman-Rubin statistic (want a value near 1)

gelman.plot(death.hier.res.c.B)

#"mixing"- effective sample size given 10,000

effectiveSize(death.hier.res.c.B[[1]][, "beta1"])

effectiveSize(death.hier.res.c.B[[1]][, "beta2"])

effectiveSize(death.hier.res.c.B[[1]][, "fi[1]"])

effectiveSize(death.hier.res.c.B[[1]][, "fi[2]"])

effectiveSize(death.hier.res.c.B[[1]][, "fi[3]"])

#ACF plot

{

autocorr.plot(death.hier.res.c.B[[1]][, "beta1"], main = "Slope1")

autocorr.plot(death.hier.res.c.B[[1]][, "beta2"], main = "Slope2")

autocorr.plot(death.hier.res.c.B[[1]][,"fi[1]"], main = "random effect 1")

autocorr.plot(death.hier.res.c.B[[1]][,"fi[2]"], main = "random effect 2")

autocorr.plot(death.hier.res.c.B[[1]][,"fi[3]"], main = "random effect 3")

}

#look at MCMC summary

summary(death.hier.res.c.B)

death.hier.c = as.data.frame(combine.mcmc(death.hier.res.c.B))

#dic.1c = dic.samples(model=death.hier.res.c.B,n.iter=10000,type="pD")

```

#2-(d)

```{r}

invlogit <- function(x){1/(1+exp(-x))}

#1,1,2015

linear.comb.2c.1 = death.hier.c$`fi[1]` + death.hier.c$beta1\*(2015 - mean(Avalanches\_part2$Season)) + death.hier.c$beta2\*(7.55 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.1 = invlogit(linear.comb.2c.1)

prob.greater.60.1 = length(which(proportion.deaths.post.expec.1>0.6))/length(proportion.deaths.post.expec.1)#0.3736333

expec1 = mean(proportion.deaths.post.expec.1)#0.3736608

ci.d.1 = quantile(proportion.deaths.post.expec.1,c(0.025,0.975))

#1,1,2018

linear.comb.2c.2 = death.hier.c$`fi[1]` + death.hier.c$beta1\*(2018 -mean(Avalanches\_part2$Season)) + death.hier.c$beta2\*(7.42 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.2 = invlogit(linear.comb.2c.2)

prob.greater.60.2 = length(which(proportion.deaths.post.expec.2>0.6))/length(proportion.deaths.post.expec.2)#0.496

expec2 = mean(proportion.deaths.post.expec.2)#0.5023339

ci.d.2 = quantile(proportion.deaths.post.expec.2,c(0.025,0.975))

#8,2,2015

linear.comb.2c.3 = death.hier.c$`fi[2]` + death.hier.c$beta1\*(2015 - mean(Avalanches\_part2$Season)) + death.hier.c$beta2\*(3.28 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.3 = invlogit(linear.comb.2c.3)

prob.greater.60.3 = length(which(proportion.deaths.post.expec.3>0.6))/length(proportion.deaths.post.expec.3)#0.3737333

expec3 = mean(proportion.deaths.post.expec.3)# 0.5001747

ci.d.3 = quantile(proportion.deaths.post.expec.3,c(0.025,0.975))

#8,2,2018

linear.comb.2c.4 = death.hier.c$`fi[2]` + death.hier.c$beta1\*(2018 - mean(Avalanches\_part2$Season)) + death.hier.c$beta2\*(6.05 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.4 = invlogit(linear.comb.2c.4)

prob.greater.60.4 = length(which(proportion.deaths.post.expec.4>0.6))/length(proportion.deaths.post.expec.4)#0.3738333

expec4 = mean(proportion.deaths.post.expec.4)#0.5030223

ci.d.4 = quantile(proportion.deaths.post.expec.4,c(0.025,0.975))

#10,3,2015

linear.comb.2c.5 = death.hier.c$`fi[3]` + death.hier.c$beta1\*(2015 - mean(Avalanches\_part2$Season)) + death.hier.c$beta2\*(2.91 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.5 = invlogit(linear.comb.2c.5)

prob.greater.60.5 = length(which(proportion.deaths.post.expec.5>0.6))/length(proportion.deaths.post.expec.5)#0.4883333

expec5 = mean(proportion.deaths.post.expec.5)# 0.5022519

ci.d.5 = quantile(proportion.deaths.post.expec.5,c(0.025,0.975))

#10,3,2018

linear.comb.2c.6 = death.hier.c$`fi[3]` + death.hier.c$beta1\*(2018 - mean(Avalanches\_part2$Season)) + death.hier.c$beta2\*(4.39 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.6 = invlogit(linear.comb.2c.6)

prob.greater.60.6 = length(which(proportion.deaths.post.expec.6>0.6))/length(proportion.deaths.post.expec.6)# 0.4955333

expec6 = mean(proportion.deaths.post.expec.6)#0.5043018

ci.d.6 = quantile(proportion.deaths.post.expec.6,c(0.025,0.975))

d.df = data.frame(

post.expectation = c(expec1,expec2,expec3,expec4,expec5,expec6),

prob.greater.60 = c(prob.greater.60.1,prob.greater.60.2,prob.greater.60.3,prob.greater.60.4,prob.greater.60.5,prob.greater.60.6),

CI\_lower = c(ci.d.1[1],ci.d.2[1],ci.d.3[1],ci.d.4[1],ci.d.5[1],ci.d.6[1]),

CI\_upwer = c(ci.d.1[2],ci.d.2[2],ci.d.3[2],ci.d.4[2],ci.d.5[2],ci.d.6[2]),

row.names = c("ststion1\_space1\_2015","ststion1\_space1\_2018","station8\_space2\_2015","station8\_space2\_2018","station10\_space3\_2015","station10\_space3\_2018")

)

round(d.df,3)

```

#2(e)

```{r}

# Data block #includes hyperparameters

n = length(Avalanches\_part2$Event\_ID)

death.hier.e.data <- list(n=n,Season=Avalanches\_part2$Season,

Snow\_total=Avalanches\_part2$Snow\_total,hit = Avalanches\_part2$Hit,

Rec.station=Avalanches\_part2$Rec.station,y.dead= Avalanches\_part2$Deaths,

J=max(Avalanches\_part2$Rec.station))

# Initial values

death.hier.e.inits <- function(){

list(fi=rnorm(max(Avalanches\_part2$Rec.station), 0,runif(1,0,10)),

beta1=rnorm(1,0,sqrt(10)),

beta2=rnorm(1,0,sqrt(10)))

}

deathrate.e.model <- "model{

#likelihood

for(i in 1:n) {

logit(mu[i]) <- fi[Rec.station[i]] + beta1\*(Season[i] - mean(Season[])) + beta2\*(Snow\_total[i] - mean(Snow\_total[]))

y.dead[i] ~ dbin(mu[i],hit[i])

}

#Prior for random effect

for(j in 1:J){

fi[j] ~ dnorm(0,1/pow(fi.se,2))

}

#hyperparameters and hyperprior

beta.mu.0= 0

beta.tau.0 = 0.1

fi.se ~ dunif(0,10)

#prior for beta

beta1 ~ dnorm(beta.mu.0,beta.tau.0)

beta2 ~ dnorm(beta.mu.0,beta.tau.0)

}"

death.hier.res.e.A <- jags.model(file=textConnection(deathrate.e.model), data=death.hier.e.data, inits=death.hier.e.inits, n.chains=3, quiet = TRUE)

update(death.hier.res.e.A, n.iter=2000)

death.hier.res.e.B <- coda.samples(death.hier.res.e.A,

variable.names=c("fi","beta1","beta2"), n.iter=10000)

```

```{r}

#MCMC Convergence Diagnostics

# Trace plots and density

plot(death.hier.res.e.B)

# Brooks-Gelman-Rubin statistic (want a value near 1)

gelman.plot(death.hier.res.e.B)

#"mixing"- effective sample size given 10,000

effectiveSize(death.hier.res.e.B[[1]][, "beta1"])

effectiveSize(death.hier.res.e.B[[1]][, "beta2"])

effectiveSize(death.hier.res.e.B[[1]][, "fi[1]"])

effectiveSize(death.hier.res.e.B[[1]][, "fi[2]"])

effectiveSize(death.hier.res.e.B[[1]][, "fi[3]"])

#ACF plot

{

autocorr.plot(death.hier.res.e.B[[1]][, "beta1"], main = "Slope1")

autocorr.plot(death.hier.res.e.B[[1]][, "beta2"], main = "Slope2")

autocorr.plot(death.hier.res.e.B[[1]][,"fi[1]"], main = "random effect 1")

autocorr.plot(death.hier.res.e.B[[1]][,"fi[2]"], main = "random effect 2")

autocorr.plot(death.hier.res.e.B[[1]][,"fi[3]"], main = "random effect 3")

}

#look at MCMC summary

summary(death.hier.res.e.B)

death.hier.e = as.data.frame(combine.mcmc(death.hier.res.e.B))

dic.c = dic.samples(model=death.hier.res.c.A,n.iter=2000,type="pD")

dic.e = dic.samples(model=death.hier.res.e.A,n.iter=2000,type="pD")

dic.b;dic.e

```

\*

```{r}

#1,1,2015

linear.comb.2e.1 = death.hier.e$`fi[1]` + death.hier.e$beta1 \* (2015 - mean(Avalanches\_part2$Season)) + death.hier.e$beta2 \* (7.55 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.e.1 = invlogit(linear.comb.2e.1)

prob.greater.60.e.1 = length(which(proportion.deaths.post.expec.e.1 > 0.6)) /

length(proportion.deaths.post.expec.e.1)

expec.e.1 = mean(proportion.deaths.post.expec.e.1)

ci.e.1 = quantile(proportion.deaths.post.expec.e.1, c(0.025, 0.975))

#1,1,2018

linear.comb.2e.2 = death.hier.e$`fi[1]` + death.hier.e$beta1 \* (2018 - mean(Avalanches\_part2$Season))+death.hier.e$beta2 \* (7.42 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.e.2 = invlogit(linear.comb.2e.2)

prob.greater.60.e.2 = length(which(proportion.deaths.post.expec.e.2 > 0.6)) /

length(proportion.deaths.post.expec.e.2)

expec.e.2 = mean(proportion.deaths.post.expec.e.2)

ci.e.2 =quantile(proportion.deaths.post.expec.e.2, c(0.025, 0.975))

#8,2,2015

linear.comb.2e.3 = death.hier.e$`fi[2]` + death.hier.e$beta1 \* (2015 - mean(Avalanches\_part2$Season)) + death.hier.e$beta2 \* (3.28 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.e.3 = invlogit(linear.comb.2e.3)

prob.greater.60.e.3 = length(which(proportion.deaths.post.expec.e.3 > 0.6)) /

length(proportion.deaths.post.expec.e.3)

expec.e.3 = mean(proportion.deaths.post.expec.e.3)

ci.e.3 =quantile(proportion.deaths.post.expec.e.3, c(0.025, 0.975))

#8,2,2018

linear.comb.2e.4 = death.hier.e$`fi[2]` + death.hier.e$beta1 \* (2018 - mean(Avalanches\_part2$Season)) + death.hier.e$beta2 \* (6.05 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.e.4 = invlogit(linear.comb.2e.4)

prob.greater.60.e.4 = length(which(proportion.deaths.post.expec.e.4 > 0.6)) /

length(proportion.deaths.post.expec.e.4)

expec.e.4 = mean(proportion.deaths.post.expec.e.4)

ci.e.4 =quantile(proportion.deaths.post.expec.e.4, c(0.025, 0.975))

#10,3,2015

linear.comb.2e.5 = death.hier.e$`fi[3]` + death.hier.e$beta1 \* (2015 - mean(Avalanches\_part2$Season)) + death.hier.e$beta2 \* (2.91 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.e.5 = invlogit(linear.comb.2e.5)

prob.greater.60.e.5 = length(which(proportion.deaths.post.expec.e.5 > 0.6)) /

length(proportion.deaths.post.expec.e.5)

expec.e.5 = mean(proportion.deaths.post.expec.e.5)

ci.e.5 =quantile(proportion.deaths.post.expec.e.5, c(0.025, 0.975))

#10,3,2018

linear.comb.2e.6 = death.hier.e$`fi[3]` + death.hier.e$beta1 \* (2018 - mean(Avalanches\_part2$Season)) + death.hier.e$beta2 \* (4.39 - mean(Avalanches\_part2$Snow\_total))

proportion.deaths.post.expec.e.6 = invlogit(linear.comb.2e.6)

prob.greater.60.e.6 = length(which(proportion.deaths.post.expec.e.6 > 0.6)) /

length(proportion.deaths.post.expec.e.6)

ci.e.6 =quantile(proportion.deaths.post.expec.e.6, c(0.025, 0.975))

e.df = data.frame(

post.expectation = c(expec.e.1,expec.e.2,expec.e.3,expec.e.4,expec.e.5,expec.e.6),

CI\_lower = c(ci.e.1[1],ci.e.2[1],ci.e.3[1],ci.e.4[1],ci.e.5[1],ci.e.6[1]),

CI\_upwer = c(ci.e.1[2],ci.e.2[2],ci.e.3[2],ci.e.4[2],ci.e.5[2],ci.e.6[2]),

row.names = c("ststion1\_space1\_2015","ststion1\_space1\_2018","station8\_space2\_2015","station8\_space2\_2018","station10\_space3\_2015","station10\_space3\_2018")

)

e.df

dic.2c<- dic.samples(model=death.hier.res.c.A,n.iter=10000,type="pD")

dic.2e<- dic.samples(model=death.hier.res.e.A,n.iter=10000,type="pD")

diffdic(dic.2c,dic.2e)#positive, the latter one is prefered

```