```
>     ##############################################################################################
>     ##############################################################################################
>     ### This file replicates Table 2
>     ##############################################################################################
>     ##############################################################################################
>
>     ### Required Adjustments
>     setwd("C:\\temp1")  # Set your working directory. All data and bugs files should be stored in this directory.
>     bugs_directory <- "C:\\WinBUGS14"  # It should be the directory of your WinBUGS program file
>
>     ### Required packages
>     library(R2WinBUGS)
Loading required package: coda
Loading required package: boot
>     library(coda)
>
>     # To create a table with results
>     bayes.easy.write <- function(id="test", stats, label, D=3){
+     M <- nrow(stats)
+     m2.sum <- rep("", 2*M)
+     m2.row <- rep("", 2*M)
+     for(i in 1:M){
+     m2.sum[2*i-1] <- paste(round(stats[i,1],digits=D), "", sep="")
+     if(max(stats[i,6],stats[i,7]) > 0 & (min(stats[i,6],stats[i,7]) > 0)) m2.sum[2*i-1] <-
paste(round(stats[i,1],digits=D), "*", sep="")
+     if(max(stats[i,6],stats[i,7]) < 0 & (min(stats[i,6],stats[i,7]) < 0)) m2.sum[2*i-1] <-
paste(round(stats[i,1],digits=D), "*", sep="")
+     if(max(stats[i,3],stats[i,4]) > 0 & (min(stats[i,3],stats[i,4]) > 0)) m2.sum[2*i-1] <-
paste(round(stats[i,1],digits=D), "**", sep="")
```

```
+     if(max(stats[i,3],stats[i,4]) < 0 & (min(stats[i,3],stats[i,4]) < 0)) m2.sum[2*i-1] <-
paste(round(stats[i,1],digits=D), "**", sep="")
+     if(max(stats[i,8],stats[i,9]) > 0 & (min(stats[i,8],stats[i,9]) > 0)) m2.sum[2*i-1] <-
paste(round(stats[i,1],digits=D), "***", sep="")
+     if(max(stats[i,8],stats[i,9]) < 0 & (min(stats[i,8],stats[i,9]) < 0)) m2.sum[2*i-1] <-
paste(round(stats[i,1],digits=D), "***", sep="")
+     m2.sum[2*i] <- paste("{", round(stats[i,2], digits=D), "}", sep="")
+     m2.row[2*i-1] <- label[i]
+     }
+     m2.table <- cbind(m2.row, m2.sum)
+     colnames(m2.table) <- c("Variable", "Coefficient")
+     return(m2.table)
+     }
>
>
>     ############################################################
>     ### House Results - First Column
>     ############################################################
>
>     # Import the individual-level data
>     Micro_Data <- read.csv("House_Data_Table2.csv", header=T)
>
>     ### Get the Congress-level data
>     Macro_Data <- read.csv("Macro_Data.csv", header=T)
>
>     T1 <- 92
>     T2 <- 114
>
>     year.odd <- seq(from=1971, to=2015, by=2)
```

```
>     T <- length(year.odd)  # number of congresses
>
>     ### Setup data
>
>     Y  <- Micro_Data$hawkishness
>
>     Party <- ifelse(Micro_Data$party==100, 1, 2)
>     GOP <- ifelse(Micro_Data$party==200, 1, 0)
>     Dem <- ifelse(Micro_Data$party==100, 1, 0)
>     Southern.State <- c(40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 51, 54) # VA, AL, AR (Arkansas), FL, GA, LA, MS, NC,
SC, TX, plus KY and TN
>     South <- ifelse(is.element(Micro_Data$ICPSR_state, Southern.State), 1, 0)
>
>     seniority <- Micro_Data$seniority
>     seniority.std  <- as.vector((seniority - mean(seniority, na.rm=T))/(sd(seniority, na.rm=T)))
>     Bases <- Micro_Data$Bases
>     Bases.std  <- as.vector((Bases - mean(Bases, na.rm=T))/(sd(Bases, na.rm=T)))
>
>     X1 <- Micro_Data$AS_cmt
>     X2 <- GOP
>     X3 <- South
>     X4 <- seniority.std
>     X5 <- Micro_Data$FR_cmt
>     X6 <- Micro_Data$veteran
>     X7 <- Bases.std
>
>     congress <- Micro_Data$congress - 91
>
```

```
>     Macro_Data$GOP_Majority <- ifelse(Macro_Data$House_Majority==200, 1, 0)
>     Macro_Data$Dem.prez <- ifelse(Macro_Data$President_party==100, 1, 0)
>
>     Z1 <- Macro_Data$GOP_Majority
>     Z2 <- Macro_Data$War
>     Z3 <- Macro_Data$Dem.prez
>
>     N <- length(Y)
>     J <- length(Z1)
>     n.beta <- 7
>     n.gamma <- 4
>
>     data <- list("N", "J", "n.beta","n.gamma", "Y",
+                  "X1","X2","X3","X4","X5","X6","X7","congress","Z1","Z2","Z3")
>
>     # setup initial values
>     init1 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), sigma.alpha=runif(1))
>     init2 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), sigma.alpha=runif(1))
>     init3 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), sigma.alpha=runif(1))
>
>     inits <- list(init1, init2, init3)
>
```

```
>     # setup parameters
>     parameters = c("beta", "sigma.y", "gamma", "alpha", "sigma.alpha")
>
>     # Test bugs.
>     Model.fit = bugs(data, inits, parameters, "table2.bug",
+                         working.directory=getwd(), bugs.directory=bugs_directory,
+                         n.chains=3, n.thin=10, n.burnin=2000, n.iter=12000, debug=F)
>
>     Rhats <- Model.fit$summary[,8]
>     range(Rhats)
[1] 1.000525 1.003742
>     # Analyze the draws from the Posterior distribution
>
>     library(coda)
>     chain1 <- read.coda(output.file="coda1.txt", index.file="codaIndex.txt", quiet=T)
>     chain2 <- read.coda(output.file="coda2.txt", index.file="codaIndex.txt", quiet=T)
>     chain3 <- read.coda(output.file="coda3.txt", index.file="codaIndex.txt", quiet=T)
>
>     MCMC <- rbind(chain1, chain2, chain3)
>     iter <- nrow(MCMC)
>
>     alpha.finder  <- seq(from=1, to=T, by=1)
>     beta.finder   <- seq(from=(T+1), to=(T+n.beta), by=1)
>     gamma.finder  <- seq(from=(T+n.beta+2), to=(T+n.beta+1+n.gamma), by=1)
>
>     alpha.mc  <- MCMC[,alpha.finder]
>     beta.mc   <- MCMC[,beta.finder]
>     gamma.mc  <- MCMC[,gamma.finder]
```

```
>
>     beta.label  <- c("Armed Services",
+                      "GOP",
+                      "South",
+                      "Seniority",
+                      "Foreign Affairs",
+                      "Veteran",
+                      "No of Military Bases")
>     gamma.label <- c("Intercept",
+                      "GOP Majority",
+                      "Major Wars",
+                      "Dem President")
>
>     # Credible Intervals
>
>     beta.CI  <- round(apply(beta.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>     gamma.CI  <- round(apply(gamma.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>     alpha.CI  <- round(apply(alpha.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>
>     beta <- round(rbind(apply(beta.mc,  2, FUN=mean), apply(beta.mc,  2, FUN=sd),    beta.CI),3)
>     rownames(beta)[c(1,2)] <- c("mean", "sd")
>     beta.stats <- t(beta)
>
>     gamma <- round(rbind(apply(gamma.mc,  2, FUN=mean), apply(gamma.mc,  2, FUN=sd),    gamma.CI),3)
>     rownames(gamma)[c(1,2)] <- c("mean", "sd")
>     gamma.stats <- t(gamma)
>
>     # Report the results
```

```
>     micro <- bayes.easy.write(id="micro_model", stats=beta.stats, label=beta.label)
>     macro <- bayes.easy.write(id="macro_model", stats=gamma.stats, label=gamma.label)
>     print(rbind(micro, macro))
      Variable               Coefficient
 [1,] "Armed Services"       "0.526***"
 [2,] ""                     "{0.037}"
 [3,] "GOP"                  "3.375***"
 [4,] ""                     "{0.024}"
 [5,] "South"                "1.1***"
 [6,] ""                     "{0.026}"
 [7,] "Seniority"            "-0.039***"
 [8,] ""                     "{0.012}"
 [9,] "Foreign Affairs"      "-0.275***"
[10,] ""                     "{0.039}"
[11,] "Veteran"              "0.056**"
[12,] ""                     "{0.026}"
[13,] "No of Military Bases" "0.049***"
[14,] ""                     "{0.012}"
[15,] "Intercept"            "-1.786***"
[16,] ""                     "{0.06}"
[17,] "GOP Majority"         "-0.299***"
[18,] ""                     "{0.066}"
[19,] "Major Wars"           "-0.077"
[20,] ""                     "{0.072}"
[21,] "Dem President"        "-0.071"
[22,] ""                     "{0.076}"
>     print(Model.fit$DIC)
[1] 32162.1
```

```
>     print(N)
[1] 10155
>


>
>     ############################################################
>     ### House Results - Second Column
>     ############################################################
>
>     # Set up data for WinBUGS
>     X1 <- Micro_Data$AS_cmt
>     X2 <- seniority.std
>     X3 <- Micro_Data$FR_cmt
>     X4 <- Bases.std
>
>     congress <- Micro_Data$congress - 91
>
>     Macro_Data$GOP_Majority <- ifelse(Macro_Data$House_Majority==200, 1, 0)
>     Macro_Data$Dem.prez <- ifelse(Macro_Data$President_party==100, 1, 0)
>
>     Z1 <- Macro_Data$GOP_Majority
>     Z2 <- Macro_Data$War
>     Z3 <- Macro_Data$Dem.prez
>
>     # create member fixed effects
>     ICPSR_ID <- as.vector(Micro_Data$ICPSR_ID)
>     uniq.icpsr_id <- unique(ICPSR_ID)
>     n.member <- length(uniq.icpsr_id)
```

```
>     member <- rep(NA, n.member)
>         for (i in 1:n.member){
+             member[ICPSR_ID == uniq.icpsr_id[i]] = i
+         }
>
>     N <- length(Y)
>     J <- length(Z1)
>     n.beta <- 4
>     n.gamma <- 4
>
>     data <- list("N", "J", "n.beta","n.gamma", "Y", "n.member",
+                  "X1","X2","X3","X4","congress","member", "Z1","Z2","Z3")
>
>     # setup initial values
>     init1 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), alpha1=rnorm(n.member), sigma.alpha=runif(1))
>     init2 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), alpha1=rnorm(n.member), sigma.alpha=runif(1))
>     init3 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), alpha1=rnorm(n.member), sigma.alpha=runif(1))
>
>     inits <- list(init1, init2, init3)
>
>     # setup parameters
>     parameters = c("beta", "sigma.y", "gamma", "alpha", "alpha1", "sigma.alpha")
```

```
>
>     # Test bugs.
>     Model.fit = bugs(data, inits, parameters, "table2_fixed.bug",
+                         working.directory=getwd(), bugs.directory=bugs_directory,
+                         n.chains=3, n.thin=5, n.burnin=1000, n.iter=6000, debug=F)
>
>     Rhats <- Model.fit$summary[,8]
>     range(Rhats)
[1] 1.000000 1.076844
>
>     # Analyze the draws from the Posterior distribution
>     chain1 <- read.coda(output.file="coda1.txt", index.file="codaIndex.txt", quiet=T)
>     chain2 <- read.coda(output.file="coda2.txt", index.file="codaIndex.txt", quiet=T)
>     chain3 <- read.coda(output.file="coda3.txt", index.file="codaIndex.txt", quiet=T)
>
>     MCMC <- rbind(chain1, chain2, chain3)
>     iter <- nrow(MCMC)
>
>     alpha.finder  <- seq(from=1, to=T, by=1)
>     alpha1.finder <- seq(from=(T+1), to=(T+n.member), by=1)
>     beta.finder   <- seq(from=(T+n.member+1), to=(T+n.member+n.beta), by=1)
>     gamma.finder  <- seq(from=(T+n.member+n.beta+2), to=(T+n.member+n.beta+1+n.gamma), by=1)
>
>     alpha.mc  <- MCMC[,alpha.finder]
>     beta.mc   <- MCMC[,beta.finder]
>     gamma.mc  <- MCMC[,gamma.finder]
>
>     beta.label  <- c("Armed Services",
```

```
+                     "Seniority",
+                     "Foreign Affairs",
+                     "No of Military Bases")
>     gamma.label <- c("Intercept",
+                     "GOP Majority",
+                     "Major Wars",
+                     "Dem President")
>
>     ### Credible Intervals
>     beta.CI  <- round(apply(beta.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>     gamma.CI  <- round(apply(gamma.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>     alpha.CI  <- round(apply(alpha.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>
>     beta <- round(rbind(apply(beta.mc,  2, FUN=mean), apply(beta.mc,  2, FUN=sd),    beta.CI),3)
>     rownames(beta)[c(1,2)] <- c("mean", "sd")
>     beta.stats <- t(beta)
>     gamma <- round(rbind(apply(gamma.mc,  2, FUN=mean), apply(gamma.mc,  2, FUN=sd),    gamma.CI),3)
>     rownames(gamma)[c(1,2)] <- c("mean", "sd")
>     gamma.stats <- t(gamma)
>
>     # summary should be like beta.stats, as above.
>     micro <- bayes.easy.write(id="micro_model", stats=beta.stats, label=beta.label)
>     macro <- bayes.easy.write(id="macro_model", stats=gamma.stats, label=gamma.label)
>     print(rbind(micro, macro))
     Variable              Coefficient
 [1,] "Armed Services"      "0.026"
 [2,] ""                    "{0.037}"
 [3,] "Seniority"           "-0.096***"
```

```
 [4,] ""                    "{0.014}"
 [5,] "Foreign Affairs"     "0.008"
 [6,] ""                    "{0.034}"
 [7,] "No of Military Bases" "0.022*"
 [8,] ""                    "{0.011}"
 [9,] "Intercept"           "0.135*"
[10,] ""                    "{0.073}"
[11,] "GOP Majority"        "-0.047"
[12,] ""                    "{0.044}"
[13,] "Major Wars"          "0.005"
[14,] ""                    "{0.042}"
[15,] "Dem President"       "0.003"
[16,] ""                    "{0.044}"
>     print(Model.fit$DIC)
[1] 17418.6
>     print(N)
[1] 10155
>
>     ############################################################
>     ### Senate Results - Third Column
>     ############################################################
>
>     # Import the individual-level data
>     Micro_Data <- read.csv("Senate_Data_Table2.csv", header=T)
>
>     # Get the Congress-level data
>     Macro_Data <- read.csv("Macro_Data.csv", header=T)
>
```

```
>     # Set up data for WinBUGS
>     Y  <- Micro_Data$hawkishness
>     Party <- ifelse(Micro_Data$party==100, 1, 2)
>     GOP <- ifelse(Micro_Data$party==200, 1, 0)
>     Dem <- ifelse(Micro_Data$party==100, 1, 0)
>     South <- ifelse(is.element(Micro_Data$ICPSR_state, Southern.State), 1, 0)
>
>     seniority <- Micro_Data$seniority
>     seniority.std  <- as.vector((seniority - mean(seniority, na.rm=T))/(sd(seniority, na.rm=T)))
>     Bases <- Micro_Data$Bases
>     Bases.std  <- as.vector((Bases - mean(Bases, na.rm=T))/(sd(Bases, na.rm=T)))
>
>     X1 <- Micro_Data$AS_cmt
>     X2 <- GOP
>     X3 <- South
>     X4 <- seniority.std
>     X5 <- Micro_Data$FR_cmt
>     X6 <- Micro_Data$veteran
>     X7 <- Bases.std
>
>     congress <- Micro_Data$congress - 91
>
>     Macro_Data$GOP_Majority <- ifelse(Macro_Data$Senate_Majority==200, 1, 0)
>     Macro_Data$Dem.prez <- ifelse(Macro_Data$President_party==100, 1, 0)
>
>     Z1 <- Macro_Data$GOP_Majority
>     Z2 <- Macro_Data$War
>     Z3 <- Macro_Data$Dem.prez
```

```
>
>     N <- length(Y)
>     J <- length(Z1)
>     n.beta <- 7
>     n.gamma <- 4
>
>     data <- list("N", "J", "n.beta","n.gamma", "Y",
+                  "X1","X2","X3","X4","X5","X6","X7","congress","Z1","Z2","Z3")
>
>     # setup initial values
>     init1 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), sigma.alpha=runif(1))
>     init2 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), sigma.alpha=runif(1))
>     init3 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), sigma.alpha=runif(1))
>
>     inits <- list(init1, init2, init3)
>
>     # setup parameters
>     parameters = c("beta", "sigma.y", "gamma", "alpha", "sigma.alpha")
>
>     # Test bugs.
>     Model.fit = bugs(data, inits, parameters, "table2.bug",
+                      working.directory=getwd(), bugs.directory=bugs_directory,
```

```
+                        n.chains=3, n.thin=10, n.burnin=2000, n.iter=12000, debug=F)
>
>     Rhats <- Model.fit$summary[,8]
>     range(Rhats)
[1] 1.000518 1.002847
>
 >     # Analyze the draws from the Posterior distribution
>     chain1 <- read.coda(output.file="coda1.txt", index.file="codaIndex.txt", quiet=T)
>     chain2 <- read.coda(output.file="coda2.txt", index.file="codaIndex.txt", quiet=T)
>     chain3 <- read.coda(output.file="coda3.txt", index.file="codaIndex.txt", quiet=T)
>
>     MCMC <- rbind(chain1, chain2, chain3)
>     iter <- nrow(MCMC)
>
>     alpha.finder  <- seq(from=1, to=T, by=1)
>     beta.finder   <- seq(from=(T+1), to=(T+n.beta), by=1)
>     gamma.finder  <- seq(from=(T+n.beta+2), to=(T+n.beta+1+n.gamma), by=1)
>
>     alpha.mc  <- MCMC[,alpha.finder]
>     beta.mc   <- MCMC[,beta.finder]
>     gamma.mc  <- MCMC[,gamma.finder]
>
>     beta.label  <- c("Armed Services",
+                      "GOP",
+                      "South",
+                      "Seniority",
+                      "Foreign Affairs",
+                      "Veteran",
```

```
+                     "No of Military Bases")
>     gamma.label <- c("Intercept",
+                     "GOP Majority",
+                     "Major Wars",
+                     "Dem President")
>
>     # Credible Intervals
>
>     beta.CI  <- round(apply(beta.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>     gamma.CI  <- round(apply(gamma.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>     alpha.CI  <- round(apply(alpha.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>
>     beta <- round(rbind(apply(beta.mc,  2, FUN=mean), apply(beta.mc,  2, FUN=sd),    beta.CI),3)
>     rownames(beta)[c(1,2)] <- c("mean", "sd")
>     beta.stats <- t(beta)
>
>     gamma <- round(rbind(apply(gamma.mc,  2, FUN=mean), apply(gamma.mc,  2, FUN=sd),    gamma.CI),3)
>     rownames(gamma)[c(1,2)] <- c("mean", "sd")
>     gamma.stats <- t(gamma)
>
>     # Report the results
>     micro <- bayes.easy.write(id="micro_model", stats=beta.stats, label=beta.label)
>     macro <- bayes.easy.write(id="macro_model", stats=gamma.stats, label=gamma.label)
>     print(rbind(micro, macro))
      Variable              Coefficient
 [1,] "Armed Services"      "0.176***"
 [2,] ""                    "{0.042}"
 [3,] "GOP"                 "2.036***"
```

```
 [4,] ""                    "{0.034}"
 [5,] "South"               "0.7***"
 [6,] ""                    "{0.041}"
 [7,] "Seniority"           "-0.011"
 [8,] ""                    "{0.017}"
 [9,] "Foreign Affairs"     "-0.19***"
[10,] ""                    "{0.043}"
[11,] "Veteran"             "-0.062*"
[12,] ""                    "{0.037}"
[13,] "No of Military Bases" "0"
[14,] ""                    "{0.018}"
[15,] "Intercept"           "-1.088***"
[16,] ""                    "{0.127}"
[17,] "GOP Majority"        "0.016"
[18,] ""                    "{0.106}"
[19,] "Major Wars"          "0.082"
[20,] ""                    "{0.128}"
[21,] "Dem President"       "-0.155"
[22,] ""                    "{0.123}"
>     print(Model.fit$DIC)
[1] 5548.07
>     print(N)
[1] 2310
>

>     ############################################################
>     ### Senate Results - Forth Column
>     ############################################################
```

```
>
>     # Set up data for WinBUGS
>     X1 <- Micro_Data$AS_cmt
>     X2 <- seniority.std
>     X3 <- Micro_Data$FR_cmt
>     X4 <- Bases.std
>
>     congress <- Micro_Data$congress - 91
>
>     Macro_Data$GOP_Majority <- ifelse(Macro_Data$Senate_Majority==200, 1, 0)
>     Macro_Data$Dem.prez <- ifelse(Macro_Data$President_party==100, 1, 0)
>
>     Z1 <- Macro_Data$GOP_Majority
>     Z2 <- Macro_Data$War
>     Z3 <- Macro_Data$Dem.prez
>
>     # create member fixed effects
>     ICPSR_ID <- as.vector(Micro_Data$ICPSR_ID)
>     uniq.icpsr_id <- unique(ICPSR_ID)
>     n.member <- length(uniq.icpsr_id)
>     member <- rep(NA, n.member)
>         for (i in 1:n.member){
+             member[ICPSR_ID == uniq.icpsr_id[i]] = i
+         }
>
>     N <- length(Y)
>     J <- length(Z1)
>     n.beta <- 4
```

```
>     n.gamma <- 4
>
>     data <- list("N", "J", "n.beta","n.gamma", "Y", "n.member",
+                  "X1","X2","X3","X4","congress","member", "Z1","Z2","Z3")
>
>     # setup initial values
>     init1 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), alpha1=rnorm(n.member), sigma.alpha=runif(1))
>     init2 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), alpha1=rnorm(n.member), sigma.alpha=runif(1))
>     init3 <- list(beta=rnorm(n.beta), sigma.y=runif(1),
+                   gamma=rnorm(n.gamma),
+                   alpha=rnorm(J), alpha1=rnorm(n.member), sigma.alpha=runif(1))
>
>     inits <- list(init1, init2, init3)
>
>     # setup parameters
>     parameters = c("beta", "sigma.y", "gamma", "alpha", "alpha1", "sigma.alpha")
>
>     # Test bugs.
>     Model.fit = bugs(data, inits, parameters, "table2_fixed.bug",
+                      working.directory=getwd(), bugs.directory=bugs_directory,
+                      n.chains=3, n.thin=10, n.burnin=2000, n.iter=12000, debug=F)
>
>     Rhats <- Model.fit$summary[,8]
>     range(Rhats)
```

```
[1] 1.000000 1.026889
>
>     # Analyze the draws from the Posterior distribution
>     chain1 <- read.coda(output.file="coda1.txt", index.file="codaIndex.txt", quiet=T)
>     chain2 <- read.coda(output.file="coda2.txt", index.file="codaIndex.txt", quiet=T)
>     chain3 <- read.coda(output.file="coda3.txt", index.file="codaIndex.txt", quiet=T)
>
>     MCMC <- rbind(chain1, chain2, chain3)
>     iter <- nrow(MCMC)
>
>     alpha.finder  <- seq(from=1, to=T, by=1)
>     alpha1.finder <- seq(from=(T+1), to=(T+n.member), by=1)
>     beta.finder   <- seq(from=(T+n.member+1), to=(T+n.member+n.beta), by=1)
>     gamma.finder  <- seq(from=(T+n.member+n.beta+2), to=(T+n.member+n.beta+1+n.gamma), by=1)
>
>     alpha.mc  <- MCMC[,alpha.finder]
>     beta.mc   <- MCMC[,beta.finder]
>     gamma.mc  <- MCMC[,gamma.finder]
>
>     beta.label  <- c("Armed Services",
+                      "Seniority",
+                      "Foreign Affairs",
+                      "No of Military Bases")
>     gamma.label <- c("Intercept",
+                      "GOP Majority",
+                      "Major Wars",
+                      "Dem President")
>
```

```
>     ### Credible Intervals
>     beta.CI  <- round(apply(beta.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>     gamma.CI  <- round(apply(gamma.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>     alpha.CI  <- round(apply(alpha.mc,  2, FUN=quantile, probs=c(0.025, 0.975, 0.5, 0.05, 0.95, 0.005, 0.995)), 3)
>
>     beta <- round(rbind(apply(beta.mc,  2, FUN=mean), apply(beta.mc,  2, FUN=sd),    beta.CI),3)
>     rownames(beta)[c(1,2)] <- c("mean", "sd")
>     beta.stats <- t(beta)
>     gamma <- round(rbind(apply(gamma.mc,  2, FUN=mean), apply(gamma.mc,  2, FUN=sd),    gamma.CI),3)
>     rownames(gamma)[c(1,2)] <- c("mean", "sd")
>     gamma.stats <- t(gamma)
>
>     # summary should be like beta.stats, as above.
>     micro <- bayes.easy.write(id="micro_model", stats=beta.stats, label=beta.label)
>     macro <- bayes.easy.write(id="macro_model", stats=gamma.stats, label=gamma.label)
>     print(rbind(micro, macro))
      Variable               Coefficient
 [1,] "Armed Services"       "-0.099**"
 [2,] ""                     "{0.049}"
 [3,] "Seniority"            "-0.127***"
 [4,] ""                     "{0.025}"
 [5,] "Foreign Affairs"      "-0.151***"
 [6,] ""                     "{0.05}"
 [7,] "No of Military Bases" "0.15***"
 [8,] ""                     "{0.032}"
 [9,] "Intercept"            "-0.49***"
[10,] ""                     "{0.147}"
[11,] "GOP Majority"         "0.105"
```

```
[12,] ""                    "{0.086}"
[13,] "Major Wars"          "0.145"
[14,] ""                    "{0.1}"
[15,] "Dem President"       "-0.041"
[16,] ""                    "{0.102}"
>     print(Model.fit$DIC)
[1] 3235.09
>     print(N)
[1] 2310
>
>
>
```