```
>     ##############################################################################################
>     ##############################################################################################
>     ### Table A1 - House
>     ##############################################################################################
>     ##############################################################################################
>
>     ### 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
Warning message:
package 'boot' was built under R version 4.0.5
>     library(coda)
>
>     # Import the individual-level data
>     Micro_Data <- read.csv("House_Count_Data.csv", header=T)
>
>     ### Get the Congress-level data
>     Macro_Data1 <- read.csv("Macro_Data.csv", header=T)
>     Macro_Data <- subset(Macro_Data1, Congress >= min(Micro_Data$congress) & Congress <= max(Micro_Data$congress))
>
>     ## Setup data
>
```

```
>     Y <- Micro_Data$rider1235
>
>     Micro_Data$LES.std <- as.vector((Micro_Data$effectiveness - mean(Micro_Data$effectiveness,
na.rm=T))/(sd(Micro_Data$effectiveness, na.rm=T)))
>     Micro_Data$Bases.std <- as.vector((Micro_Data$Bases - mean(Micro_Data$Bases, na.rm=T))/(sd(Micro_Data$Bases,
na.rm=T)))
>
>     X1.1 <- Micro_Data$hawkish
>     X1  <- as.vector((X1.1 - mean(X1.1, na.rm=T))/(sd(X1.1, na.rm=T)))
>     X2 <- Micro_Data$Majority
>     X3.1 <- Micro_Data$seniority
>     X3  <- as.vector((X3.1 - mean(X3.1, na.rm=T))/(sd(X3.1, na.rm=T)))
>     X4 <- Micro_Data$FR_cmt
>     X5 <- Micro_Data$AS_cmt
>     X6 <- Micro_Data$veteran
>     X7 <- Micro_Data$Bases.std
>     X8 <- Micro_Data$LES.std
>
>     W1 <- X1
>     W2 <- Micro_Data$FR_cmt
>     W3 <- Micro_Data$AS_cmt
>
>     # Standardize the continuous variables
>     Macro_Data$GOP_Majority <- ifelse(Macro_Data$House_Majority==200, 1, 0)
>     Dem.prez <- ifelse(Macro_Data$President_party==100, 1, 0)
>
>     Z1 <- Macro_Data$War
>     Z2 <- Macro_Data$GOP_Majority
>     Z3 <- Dem.prez
```

```
>
>     # create time indicators
>     cong <- as.vector(Micro_Data$congress)
>     uniq.cong <- unique(cong)
>     n.cong <- length(uniq.cong)
>     time <- rep(NA, n.cong)
>         for (i in 1:n.cong){
+             time[cong == uniq.cong[i]] = i
+         }
>
>     congress <- time
>
>     N <- length(Y)
>     J <- length(Z1)
>     n.beta <- 8
>     n.gamma <- 4
>     n.delta <- 4
>
>     data <- list("N", "J", "n.beta", "n.delta", "n.gamma", "Y", "W1", "W2", "W3",
+                  "X1","X2","X3","X4","X5","X6","X7","X8", "congress",
+                  "Z1","Z2","Z3")
>
>     # setup initial values
>     # Note I provide initial values of u to make sure u is an integer.
>     init1 <- list(u=sample(c(0,1), N, replace=T), gamma=rnorm(n.gamma), beta=rnorm(n.beta),
+                   delta=rnorm(n.delta), alpha=rnorm(J))
>     init2 <- list(u=sample(c(0,1), N, replace=T), gamma=rnorm(n.gamma), beta=rnorm(n.beta),
+                   delta=rnorm(n.delta), alpha=rnorm(J))
```

```
>     init3 <- list(u=sample(c(0,1), N, replace=T), gamma=rnorm(n.gamma), beta=rnorm(n.beta),
+                   delta=rnorm(n.delta), alpha=rnorm(J))
>
>     inits <- list(init1, init2, init3)
>
>     # setup parameters
>     parameters = c("beta", "delta", "gamma", "alpha")
>         # Test bugs.
>    Model.fit = bugs(data, inits, parameters, "table_A1.bug",
>                        working.directory=getwd(), bugs.directory=bugs_directory,
>                        n.chains=3, n.thin=1, n.burnin=500, n.iter=1500)
>
>    Rhats <- Model.fit$summary[,8]
>    range(Rhats)
 [1] 1.000987 1.117152
>         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=n.cong, by=1)
>     beta.finder <- (n.cong+1):(n.beta + n.cong)
>     delta.finder  <- seq(from=(n.beta + n.cong + 1), to=(n.beta + n.cong + n.delta), by=1)
>     gamma.finder  <- seq(from=(n.beta + n.cong + n.delta + 2), to=(n.beta + n.cong + n.delta + n.gamma + 1), by=1)
>
>     alpha.mc  <- MCMC[,alpha.finder]
```

```
>     beta.mc   <- MCMC[,beta.finder]
>     delta.mc  <- MCMC[,delta.finder]
>     gamma.mc  <- MCMC[,gamma.finder]
>     deviance.mc <- MCMC[,(n.beta + n.cong + n.delta + 1)]
>
>     beta.label <- c("Hawkishness",
+                     "Majority",
+                     "Seniority",
+                     "Foreign Affairs",
+                     "Armed Services",
+                     "Veteran",
+                     "No of Military Bases",
+                     "Effectiveness")
>
>     delta.label <- c("Intercept",
+                     "Hawkishness",
+                     "Foreign Affairs",
+                     "Armed Services")
>
>     gamma.label <- c("Intercept",
+                      "Major War",
+                      "GOP Majority",
+                      "Dem Prez"
+                      )
>
>     ### 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)
>     delta.CI  <- round(apply(delta.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")
>     delta <- round(rbind(apply(delta.mc,  2, FUN=mean), apply(delta.mc,  2, FUN=sd),    delta.CI),3)
>     rownames(delta)[c(1,2)] <- c("mean", "sd")
>     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")
>     delta.stats <- t(delta)
>     beta.stats <- t(beta)
>     gamma.stats <- t(gamma)
>
>     total.stats <- rbind(delta.stats, beta.stats, gamma.stats, mean(deviance.mc))
>     total.label <- c(delta.label, beta.label, gamma.label, "deviance")
>     row.names(total.stats) <- total.label
>
>     m2.sum <- rep("", 2*nrow(total.stats))
>     m2.row <- rep("", 2*nrow(total.stats))
>
>     for(i in 1:nrow(total.stats)){
+     if (total.stats[i,6] > 0 & total.stats[i,7] > 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "*",
sep="")
+     if (total.stats[i,6] < 0 & total.stats[i,7] < 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "*",
sep="")
+     if (total.stats[i,3] > 0 & total.stats[i,4] > 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "**",
sep="")
+     if (total.stats[i,3] < 0 & total.stats[i,4] < 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "**",
sep="")
```

```
+     if (total.stats[i,8] > 0 & total.stats[i,9] > 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "***",
sep="")
+     if (total.stats[i,8] < 0 & total.stats[i,9] < 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "***",
sep="")
+     if (total.stats[i,6] < 0 & total.stats[i,7] > 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "",
sep="")
+     if (total.stats[i,6] > 0 & total.stats[i,7] < 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "",
sep="")
+     m2.sum[2*i] <- paste("{", round(total.stats[i,2], digits=2), "}", sep="")
+     m2.row[2*i-1] <- total.label[i]
+     }
>     m2.table <- cbind(m2.row, m2.sum)
>     print(m2.table)
      m2.row                m2.sum
 [1,] "Intercept"           "2.16***"
 [2,] ""                    "{0.14}"
 [3,] "Hawkishness"         "-0.05"
 [4,] ""                    "{0.09}"
 [5,] "Foreign Affairs"     "-1.01***"
 [6,] ""                    "{0.44}"
 [7,] "Armed Services"      "-1.26***"
 [8,] ""                    "{0.22}"
 [9,] "Hawkishness"         "-0.35***"
[10,] ""                    "{0.07}"
[11,] "Majority"            "-0.18"
[12,] ""                    "{0.11}"
[13,] "Seniority"           "0.17***"
[14,] ""                    "{0.05}"
[15,] "Foreign Affairs"     "-0.4"
[16,] ""                    "{0.32}"
```

```
[17,] "Armed Services"       "0.58***"
[18,] ""                     "{0.17}"
[19,] "Veteran"              "0.01"
[20,] ""                     "{0.1}"
[21,] "No of Military Bases" "-0.1**"
[22,] ""                     "{0.05}"
[23,] "Effectiveness"        "0.11**"
[24,] ""                     "{0.05}"
[25,] "Intercept"            "-0.96**"
[26,] ""                     "{0.36}"
[27,] "Major War"            "-0.75*"
[28,] ""                     "{0.4}"
[29,] "GOP Majority"         "-0.21"
[30,] ""                     "{0.36}"
[31,] "Dem Prez"             "0.04"
[32,] ""                     "{0.42}"
[33,] "deviance"             "2015.69***"
[34,] ""                     "{2015.69}"
>     print(N)
[1] 10192
>
>
>

>     ##############################################################################################
>     ##############################################################################################
>     ### Table A1 - Senate
>     ##############################################################################################
```

```
>     ##############################################################################################
>     # Import the individual-level data
>     Micro_Data <- read.csv("Senate_Count_Data.csv", header=T)
>
>     # Congress-level data
>     Macro_Data <- read.csv("Count_Macro_Data.csv", header=T)
>
>     ### Setup data
>     Y <- Micro_Data$rider1235
>
>     ## Micro-level variables
>     Party <- ifelse(Micro_Data$party==100, 1, 2)
>     GOP <- ifelse(Micro_Data$party==200, 1, 0)
>
>     Majority <- ifelse(Micro_Data$party_code == Micro_Data$Majority, 1, 0)
>     Micro_Data$LES.std <- as.vector((Micro_Data$effectiveness - mean(Micro_Data$effectiveness,
na.rm=T))/(sd(Micro_Data$effectiveness, na.rm=T)))
>     Micro_Data$Bases.std <- as.vector((Micro_Data$Bases - mean(Micro_Data$Bases, na.rm=T))/(sd(Micro_Data$Bases,
na.rm=T)))
>
>     X1.1 <- Micro_Data$hawkish
>     X1  <- as.vector((X1.1 - mean(X1.1, na.rm=T))/(sd(X1.1, na.rm=T)))
>     X2 <- Majority
>     X3.1 <- Micro_Data$Seniority_Year
>     X3  <- as.vector((X3.1 - mean(X3.1, na.rm=T))/(sd(X3.1, na.rm=T)))
>     X4 <- Micro_Data$FR_cmt
>     X5 <- Micro_Data$AS_cmt
>     X6 <- Micro_Data$veteran
>     X7 <- Micro_Data$Bases.std
```

```
>     X8 <- Micro_Data$LES.std
>
>     W1 <- X1
>     W2 <- Micro_Data$FR_cmt
>     W3 <- Micro_Data$AS_cmt
>
>     # Standardize the continuous variables
>     Macro_Data$GOP_Majority <- ifelse(Macro_Data$Senate_Majority==200, 1, 0)
>     Dem.prez <- ifelse(Macro_Data$President_party==100, 1, 0)
>
>     Z1 <- Macro_Data$War
>     Z2 <- Macro_Data$GOP_Majority
>     Z3 <- Dem.prez
>
>     # create time indicators
>     cong <- as.vector(Micro_Data$congress)
>     uniq.cong <- unique(cong)
>     n.cong <- length(uniq.cong)
>     time <- rep(NA, n.cong)
>         for (i in 1:n.cong){
+             time[cong == uniq.cong[i]] = i
+         }
>
>     congress <- time
>
>
>     N <- length(Y)
>     J <- length(Z1)
```

```
>     n.beta <- 8
>     n.gamma <- 4
>     n.delta <- 4
>
>     data <- list("N", "J", "n.beta", "n.delta", "n.gamma", "Y", "W1", "W2", "W3",
+                  "X1","X2","X3","X4","X5","X6","X7","X8", "congress",
+                  "Z1","Z2","Z3")
>
>     # setup initial values
>     # Note I provide initial values of u to make sure u is an integer.
>     init1 <- list(u=sample(c(0,1), N, replace=T), gamma=rnorm(n.gamma), beta=rnorm(n.beta),
+                   delta=rnorm(n.delta), alpha=rnorm(J))
>     init2 <- list(u=sample(c(0,1), N, replace=T), gamma=rnorm(n.gamma), beta=rnorm(n.beta),
+                   delta=rnorm(n.delta), alpha=rnorm(J))
>     init3 <- list(u=sample(c(0,1), N, replace=T), gamma=rnorm(n.gamma), beta=rnorm(n.beta),
+                   delta=rnorm(n.delta), alpha=rnorm(J))
>
>     inits <- list(init1, init2, init3)
>
>     # setup parameters
>     parameters = c("beta", "delta", "gamma", "alpha")
>
>     # Test bugs.
>     Model.fit = bugs(data, inits, parameters, "table_A1.bug",
+                          working.directory=getwd(), bugs.directory=bugs_directory,
+                          n.chains=3, n.thin=10, n.burnin=1000, n.iter=6000)
>
>     Rhats <- Model.fit$summary[,8]
```

```
>     range(Rhats)
[1] 1.000057 1.010279
>
>     # 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=n.cong, by=1)
>     beta.finder <- (n.cong+1):(n.beta + n.cong)
>     delta.finder  <- seq(from=(n.beta + n.cong + 1), to=(n.beta + n.cong + n.delta), by=1)
>     gamma.finder  <- seq(from=(n.beta + n.cong + n.delta + 2), to=(n.beta + n.cong + n.delta + n.gamma + 1), by=1)
>
>     alpha.mc  <- MCMC[,alpha.finder]
>     beta.mc   <- MCMC[,beta.finder]
>     delta.mc  <- MCMC[,delta.finder]
>     gamma.mc  <- MCMC[,gamma.finder]
>     deviance.mc <- MCMC[,(n.beta + n.cong + n.delta + 1)]
>
>     beta.label <- c("Hawkishness",
+                     "Majority",
+                     "Seniority",
+                     "Foreign Affairs",
+                     "Armed Services",
+                     "Veteran",
```

```
+                    "No of Military Bases",
+                    "Effectiveness")
>
>     delta.label <- c("Intercept",
+                    "Hawkishness",
+                    "Foreign Affairs",
+                    "Armed Services")
>
>     gamma.label <- c("Intercept",
+                     "Major War",
+                     "GOP Majority",
+                     "Dem Prez"
+                     )
>
>     ### 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)
>     delta.CI  <- round(apply(delta.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")
>     delta <- round(rbind(apply(delta.mc,  2, FUN=mean), apply(delta.mc,  2, FUN=sd),    delta.CI),3)
>     rownames(delta)[c(1,2)] <- c("mean", "sd")
>     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")
>     delta.stats <- t(delta)
```

```
>     beta.stats <- t(beta)
>     gamma.stats <- t(gamma)
>
>     total.stats <- rbind(delta.stats, beta.stats, gamma.stats, mean(deviance.mc))
>     total.label <- c(delta.label, beta.label, gamma.label, "deviance")
>     row.names(total.stats) <- total.label
>
>     m2.sum <- rep("", 2*nrow(total.stats))
>     m2.row <- rep("", 2*nrow(total.stats))
>
>     for(i in 1:nrow(total.stats)){
+     if (total.stats[i,6] > 0 & total.stats[i,7] > 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "*",
sep="")
+     if (total.stats[i,6] < 0 & total.stats[i,7] < 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "*",
sep="")
+     if (total.stats[i,3] > 0 & total.stats[i,4] > 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "**",
sep="")
+     if (total.stats[i,3] < 0 & total.stats[i,4] < 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "**",
sep="")
+     if (total.stats[i,8] > 0 & total.stats[i,9] > 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "***",
sep="")
+     if (total.stats[i,8] < 0 & total.stats[i,9] < 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "***",
sep="")
+     if (total.stats[i,6] < 0 & total.stats[i,7] > 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "",
sep="")
+     if (total.stats[i,6] > 0 & total.stats[i,7] < 0) m2.sum[2*i-1] <- paste(round(total.stats[i,1],digits=2), "",
sep="")
+     m2.sum[2*i] <- paste("{", round(total.stats[i,2], digits=2), "}", sep="")
+     m2.row[2*i-1] <- total.label[i]
+     }
>     m2.table <- cbind(m2.row, m2.sum)
```

```
>     print(m2.table)
      m2.row                  m2.sum
 [1,] "Intercept"             "1.32***"
 [2,] ""                      "{0.14}"
 [3,] "Hawkishness"           "0.22*"
 [4,] ""                      "{0.12}"
 [5,] "Foreign Affairs"       "-0.03"
 [6,] ""                      "{0.26}"
 [7,] "Armed Services"        "-1.47***"
 [8,] ""                      "{0.24}"
 [9,] "Hawkishness"           "0"
[10,] ""                      "{0.08}"
[11,] "Majority"              "0.09"
[12,] ""                      "{0.1}"
[13,] "Seniority"             "0.19***"
[14,] ""                      "{0.05}"
[15,] "Foreign Affairs"       "-0.11"
[16,] ""                      "{0.14}"
[17,] "Armed Services"        "0.01"
[18,] ""                      "{0.11}"
[19,] "Veteran"               "0.07"
[20,] ""                      "{0.1}"
[21,] "No of Military Bases" "0.1*"
[22,] ""                      "{0.05}"
[23,] "Effectiveness"         "0.04"
[24,] ""                      "{0.05}"
[25,] "Intercept"             "-0.31"
[26,] ""                      "{0.33}"
```

```
[27,] "Major War"            "-0.27"
[28,] ""                     "{0.32}"
[29,] "GOP Majority"         "0.04"
[30,] ""                     "{0.26}"
[31,] "Dem Prez"             "-0.7**"
[32,] ""                     "{0.29}"
[33,] "deviance"             "1251.5***"
[34,] ""                     "{1251.5}"
>     print(N)
[1] 2186
>
```