Replication Report for The Four Faces of Political Participation in Argentina: Using Latent Class Analysis to Study Political Behavior

Replication

The model used MCMC and it's impossible to run it in Rmarkdown. I pushed the code and the original data to Uchicago's CPU cluster and ran it. Code and output data are as follows.

Original Data: https://github.com/yifan-lyf/40800-replication/blob/main/datamodel.RData

Stage1: https://github.com/yifan-lyf/40800-replication/blob/main/stage1.R

sbatch file for stage1: https://github.com/yifan-lyf/40800-replication/blob/main/stage1.sbatch

Stage2: https://github.com/yifan-lyf/40800-replication/blob/main/stage2.R

sbatch file for stage2: https://github.com/yifan-lyf/40800-replication/blob/main/stage2.sbatch

Data produced by stage1: https://github.com/yifan-lyf/40800-replication/blob/main/samples_stage1.Rdata

Data produced by stage2: https://github.com/yifan-lyf/40800-replication/blob/main/samples_stage2.Rdata

Descriptive Analysis

```r
library(rjags)
library(coda)
load.module("glm")

library(foreach)
library(doMC)
registerDoMC(4)

RNGkind("L'Ecuyer-CMRG")
set.seed(100)
# Load recoded data

load("datamodel.Rdata")

attach(datamodel)

set.seed(100)


n <- dim(datamodel)[1]

fixed.unconv <- rep(0, n)
fixed.unconv[unconv.scale == 3 & conv.scale == 0] <- 2
```

```
fixed.unconv[unconv.scale == 0 & conv.scale > 6] <- 1
which(fixed.unconv != 0)
```

```
 [1]  117  624  687 1808 1874 2113 2723 3181 3295 3349 3429 3446 3611 3739 4130
[16] 4147 4152
```

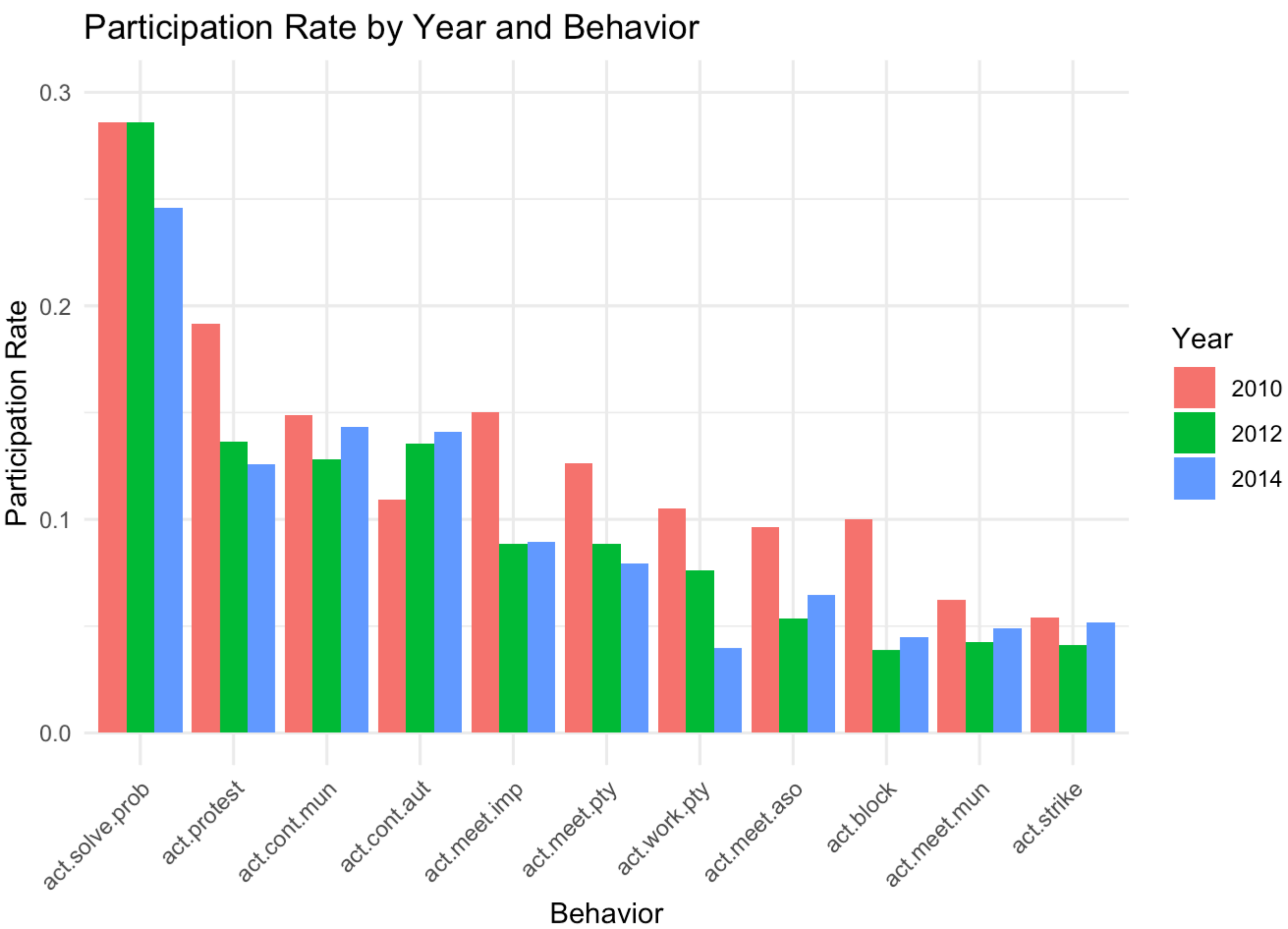
```
fixed.conv <- rep(0, n)
fixed.conv[unconv.scale == 3 & conv.scale == 0] <- 1
fixed.conv[unconv.scale == 0 & conv.scale > 6] <- 2
which(fixed.conv != 0)
```

```
 [1]  117  624  687 1808 1874 2113 2723 3181 3295 3349 3429 3446 3611 3739 4130
[16] 4147 4152
```

```
Y <- cbind(act.meet.mun, act.cont.mun, act.cont.aut, act.meet.imp, act.meet.pty, act.meet.

ntypes <- 2

nact <- dim(Y)[2]

nconv <- nact - 3

wprior <- c(1, 1)
```

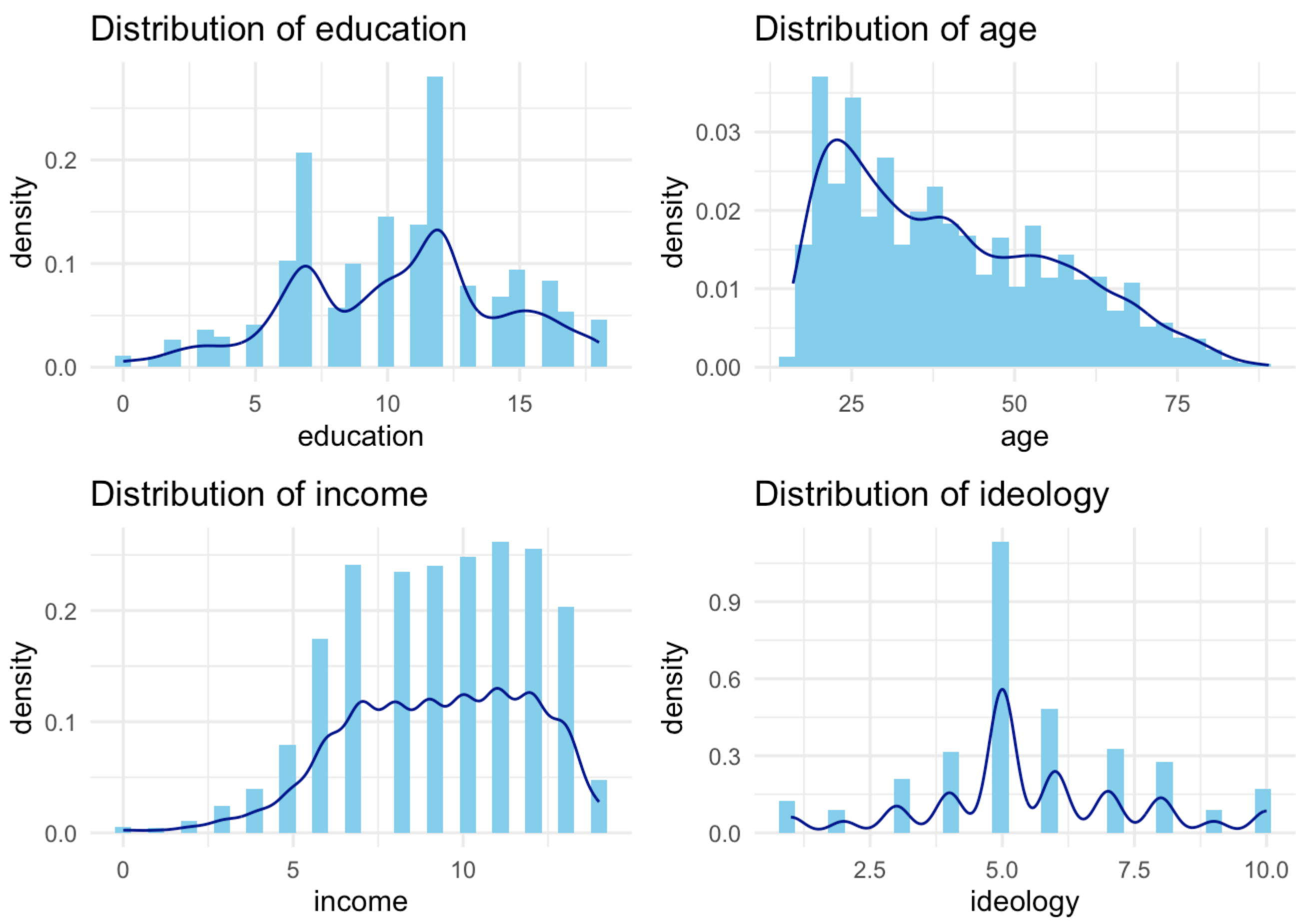
```
library(ggplot2)
library(reshape2)
library(dplyr)

Y_df <- as.data.frame(Y)
Y_df$year <- datamodel$year.cat

Y_long <- melt(Y_df, id.vars = "year", variable.name = "Behavior", value.name = "Value")

summary_df <- Y_long %>%
  group_by(year, Behavior) %>%
  summarise(ParticipationRate = mean(Value, na.rm = TRUE), .groups = "drop")

summary_df$year <- factor(summary_df$year, labels = c("2010", "2012", "2014"))

summary_df$Behavior <- factor(summary_df$Behavior,
                              levels = summary_df %>%
                                group_by(Behavior) %>%
                                summarise(avg = mean(ParticipationRate)) %>%
                                arrange(desc(avg)) %>%
                                pull(Behavior))

ggplot(summary_df, aes(x = Behavior, y = ParticipationRate, fill = year)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(title = "Participation Rate by Year and Behavior",
       x = "Behavior",
       y = "Participation Rate",
       fill = "Year") +
```

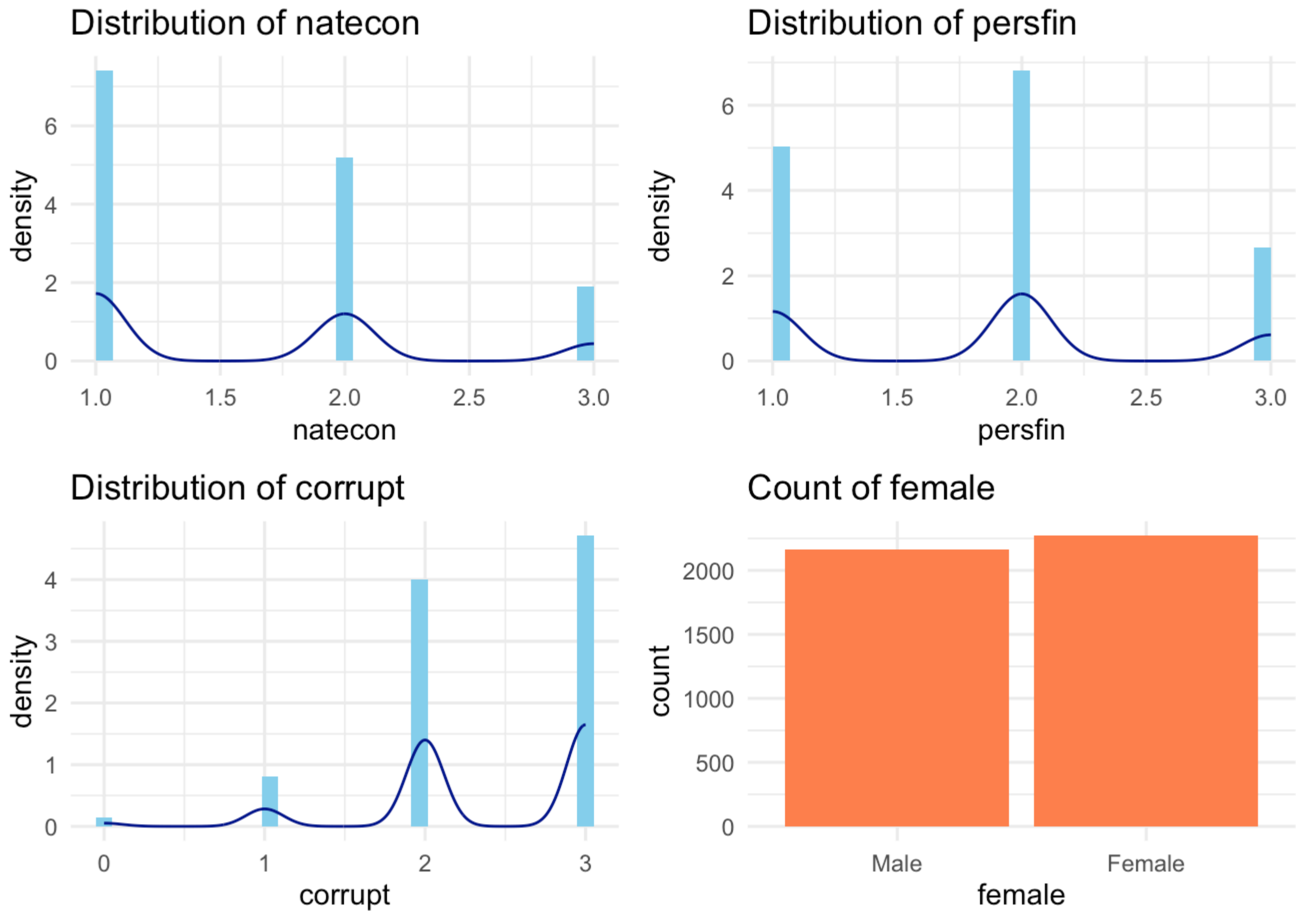
```
  ylim(0, 0.3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

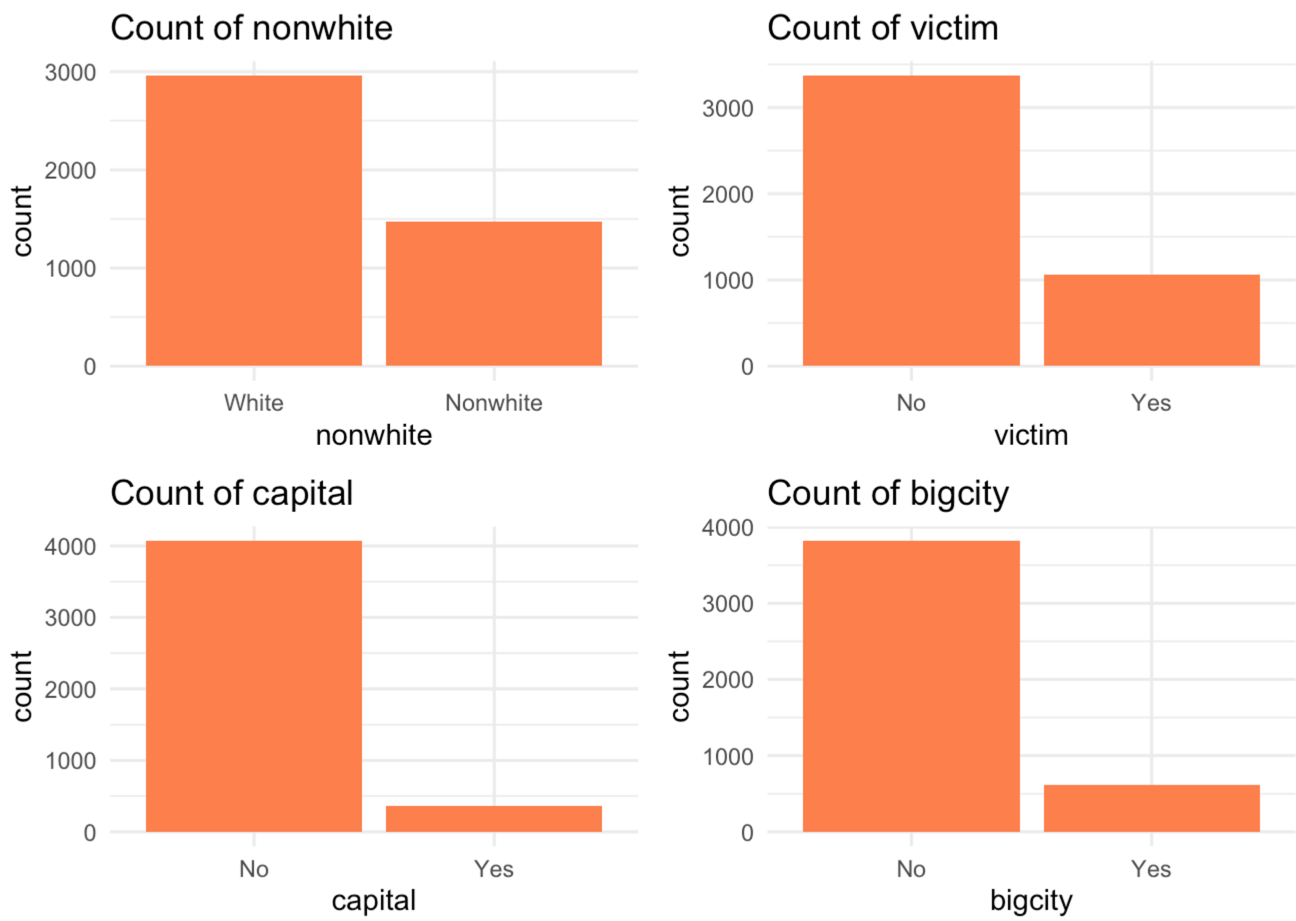


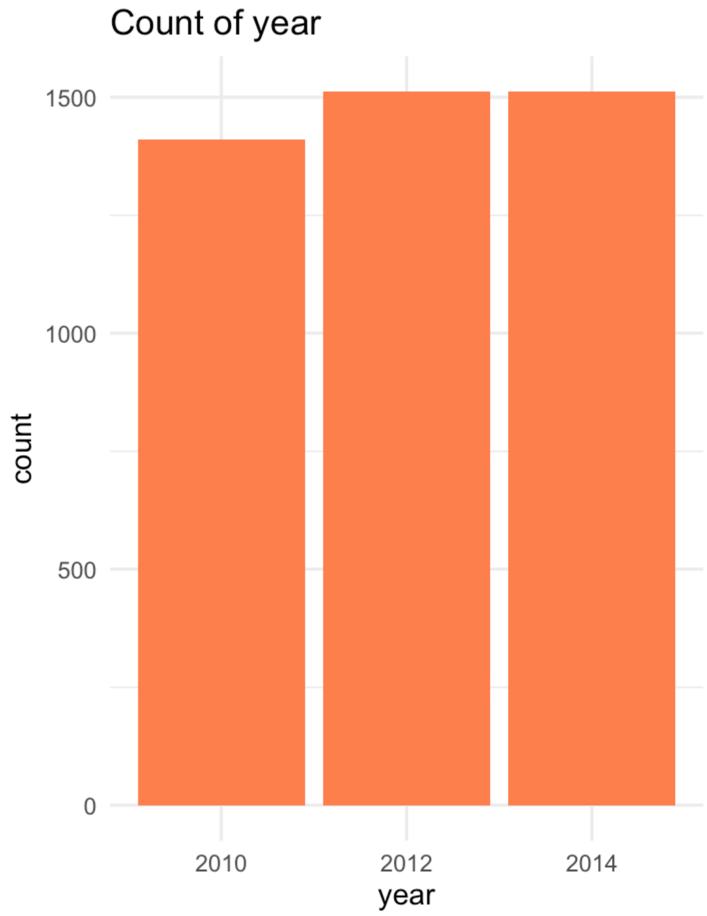

```
library(ggplot2)
library(patchwork)

numeric_vars <- c("education", "age", "income", "ideology", "natecon", "persfin", "corrupt
categorical_vars <- c("female", "nonwhite", "victim", "capital", "bigcity", "year")

df <- data.frame(
  education = datamodel$education,
  age = datamodel$age,
  female = factor(datamodel$female, labels = c("Male", "Female")),
  nonwhite = factor(datamodel$nonwhite, labels = c("White", "Nonwhite")),
  income = datamodel$income.proxy,
  ideology = datamodel$ideology,
  natecon = datamodel$natecon,
  persfin = datamodel$persfin,
  victim = factor(datamodel$victim, labels = c("No", "Yes")),
  corrupt = datamodel$corrupt,
  capital = factor(datamodel$capital, labels = c("No", "Yes")),
  bigcity = factor(datamodel$bigcity, labels = c("No", "Yes")),
  year = factor(datamodel$year.cat, labels = c("2010", "2012", "2014"))
)

make_plot <- function(var) {
  if (is.numeric(df[[var]])) {
    ggplot(df, aes_string(x = var)) +
      geom_histogram(aes(y = ..density..), fill = "skyblue", bins = 30) +
```

```
      geom_density(color = "darkblue") +
      theme_minimal() +
      ggtitle(paste("Distribution of", var))
  } else {
    ggplot(df, aes_string(x = var)) +
      geom_bar(fill = "coral") +
      theme_minimal() +
      ggtitle(paste("Count of", var))
  }
}

all_vars <- c(numeric_vars, categorical_vars)
all_plots <- lapply(all_vars, make_plot)

pages <- split(all_plots, ceiling(seq_along(all_plots)/4))

for (i in seq_along(pages)) {
  print(wrap_plots(pages[[i]], ncol = 2))
}
```




Distribution of natecon
density
natecon
Distribution of persfin
density
persfin
Distribution of corrupt
density
corrupt
Count of female
count
Male
Female
female


Count of nonwhite
count
White
Nonwhite
nonwhite
Count of victim
count
No
Yes
victim
Count of capital
count
No
Yes
capital
Count of bigcity
count
No
Yes
bigcity



Extension

Original Data: https://github.com/yifan-lyf/40800-replication/blob/main/datamodel.RData

Stage1:

https://github.com/yifan-lyf/40800-replication/blob/main/param/stage1_0.1.R

https://github.com/yifan-lyf/40800-replication/blob/main/param/stage1_0.5.R

sbatch file for stage1:

https://github.com/yifan-lyf/40800-replication/blob/main/param/stage1_0.1.sbatch

https://github.com/yifan-lyf/40800-replication/blob/main/param/stage1_0.5.sbatch

Stage2:

https://github.com/yifan-lyf/40800-replication/blob/main/param/stage2_0.1.R

https://github.com/yifan-lyf/40800-replication/blob/main/param/stage2_0.5.R

sbatch file for stage2:

https://github.com/yifan-lyf/40800-replication/blob/main/param/stage2_0.1.sbatch

https://github.com/yifan-lyf/40800-replication/blob/main/param/stage2_0.5.sbatch

Data produced by stage1:

https://github.com/yifan-lyf/40800-replication/blob/main/param/samples_stage1_0.1.Rdata

https://github.com/yifan-lyf/40800-replication/blob/main/param/samples_stage1_0.5.Rdata

Data produced by stage2:

https://github.com/yifan-lyf/40800-replication/blob/main/param/samples_stage2_0.1.Rdata

https://github.com/yifan-lyf/40800-replication/blob/main/param/samples_stage2_0.5.Rdata

Visualization

```
# Goal: Make Table 1 and Figures 1-2
# Dependencies: "samples_stage1.Rdata"

library(xtable)
library(scales)
library(gplots)

load("samples_stage1.Rdata")

options(digits = 3)
options(scipen = 999)

# We first look at the results of the latent class model, usted to classify respondents in

# (1) Low conventional, low unconventional (Outsider)
# (2) Low conventional, high unconventional (Agitator)
# (3) High conventional, low unconventional (Conventional)
# (4) High conventional, high unconventional (Activist)

## Influence of participatory types on involvement in political activities

# Our expectation is that $\alpha_{C, j}$ should be higher for conventional activities, an

# To ensure identification of model parameters, we set $\alpha_{C, j}$ to zero for the act

# The following table give average values and 95% posterior intervals for all $\alpha_{C,

act.names <- c("Municipal meetings", "Contact municipality", "Contact authorities", "Impro

alpha.samples <- rbind(samples.stage1[[1]][ , substr(colnames(samples.stage1[[1]]), 1, 6)

alpha.conv.samples <- rbind(samples.stage1[[1]][ , substr(colnames(samples.stage1[[1]]), 1

alpha.unconv.samples <- rbind(samples.stage1[[1]][ , substr(colnames(samples.stage1[[1]]),

T.ac.i <- apply(alpha.conv.samples, 2, mean)
T.ac.ii <- apply(alpha.conv.samples, 2, quantile, p = c(0.025, 0.975))
T.ac <- t(rbind(T.ac.i, T.ac.ii))
colnames(T.ac) <- c("mean", "2.5%", "97.5%")
rownames(T.ac) <- act.names

T.au.i <- apply(alpha.unconv.samples, 2, mean)
T.au.ii <- apply(alpha.unconv.samples, 2, quantile, p = c(0.025, 0.975))
```

```r
T.au <- t(rbind(T.au.i, T.au.ii))
colnames(T.au) <- c("mean", "2.5%", "97.5%")
rownames(T.au) <- act.names
```

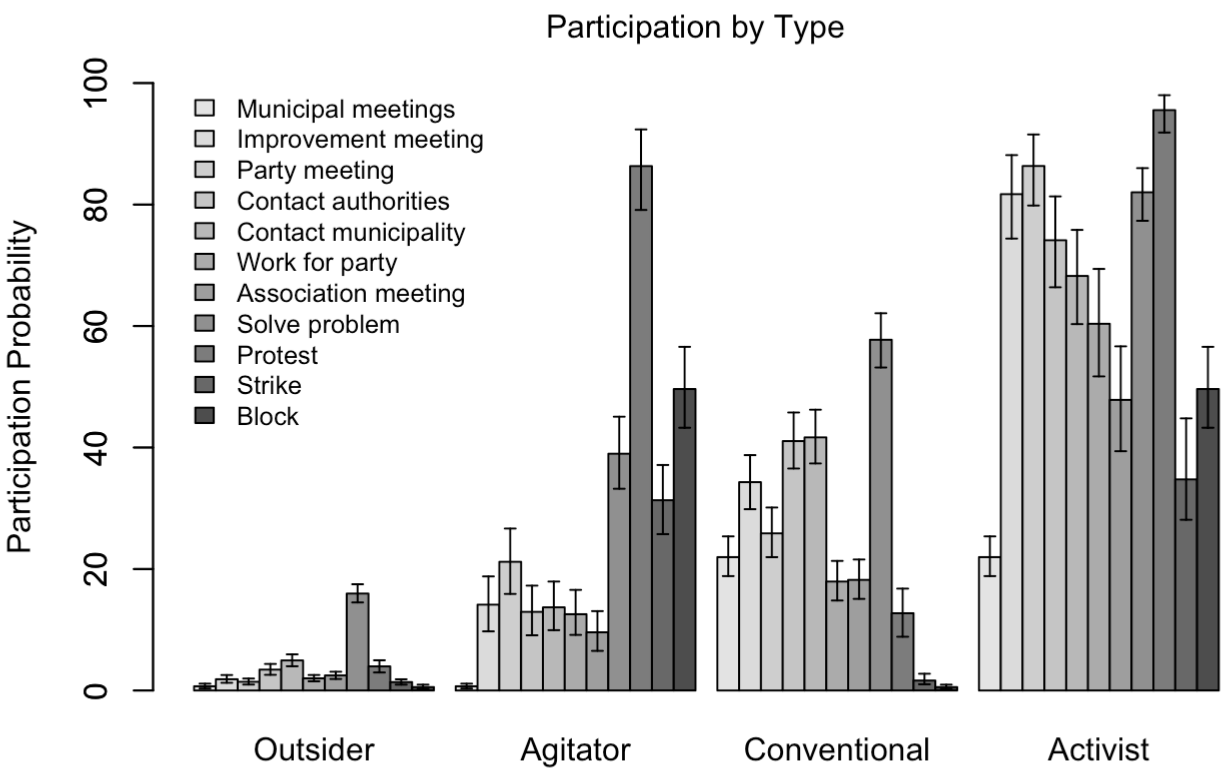
```r
Table1 <- cbind(T.ac, T.au)

xtable(Table1)
```

```
% latex table generated in R 4.4.2 by xtable 1.8-4 package
% Sat May 31 01:44:12 2025
\begin{table}[ht]
\centering
\begin{tabular}{rrrrrrr}
  \hline
 & mean & 2.5\% & 97.5\% & mean & 2.5\% & 97.5\% \\
  \hline
Municipal meetings & 3.74 & 3.16 & 4.48 & 0.00 & 0.00 & 0.00 \\
  Contact municipality & 2.62 & 2.34 & 2.90 & 1.11 & 0.77 & 1.45 \\
  Contact authorities & 2.98 & 2.66 & 3.32 & 1.43 & 1.05 & 1.82 \\
  Improvement meeting & 3.33 & 2.96 & 3.75 & 2.17 & 1.73 & 2.61 \\
  Party meeting & 3.19 & 2.77 & 3.64 & 2.92 & 2.48 & 3.40 \\
  Association meeting & 2.17 & 1.85 & 2.51 & 1.42 & 1.03 & 1.78 \\
  Solve problem & 1.98 & 1.76 & 2.19 & 1.21 & 0.94 & 1.47 \\
  Work for party & 2.38 & 2.06 & 2.72 & 1.95 & 1.59 & 2.32 \\
  Protest & 1.26 & 0.83 & 1.66 & 5.07 & 4.48 & 5.74 \\
  Strike & 0.15 & 0.00 & 0.67 & 3.49 & 3.09 & 3.90 \\
  Block & 0.00 & 0.00 & 0.00 & 5.23 & 4.58 & 6.21 \\
   \hline
\end{tabular}
\end{table}
```

```r
# As expected, $\alpha_{C, j}$'s are higher for conventional activities than unconventiona

### Type Effects

# For each combination of conventional and unconventional types, we can compute predicted

inv.logit <- function(x) {
  y <- 1 / (1 + exp(-x))
  return(y)
}

n.iters <- dim(samples.stage1[[1]])[1] * 3
n.act <- dim(alpha.samples)[2]
n.ctypes <- 4
pred.P <- array(NA, c(n.iters, n.act, n.ctypes))
pred.change.P <- array(NA, c(n.iters, n.act, (n.ctypes - 1)))

pred.P[,,1] <- inv.logit(alpha.samples)
pred.P[,,2] <- inv.logit(alpha.samples + alpha.unconv.samples)
pred.P[,,3] <- inv.logit(alpha.samples + alpha.conv.samples)
pred.P[,,4] <- inv.logit(alpha.samples + alpha.conv.samples + alpha.unconv.samples)

pred.change.P[ , , 1] <- pred.P[ , , 2] - pred.P[ , , 1]
pred.change.P[ , , 2] <- pred.P[ , , 3] - pred.P[ , , 1]
```

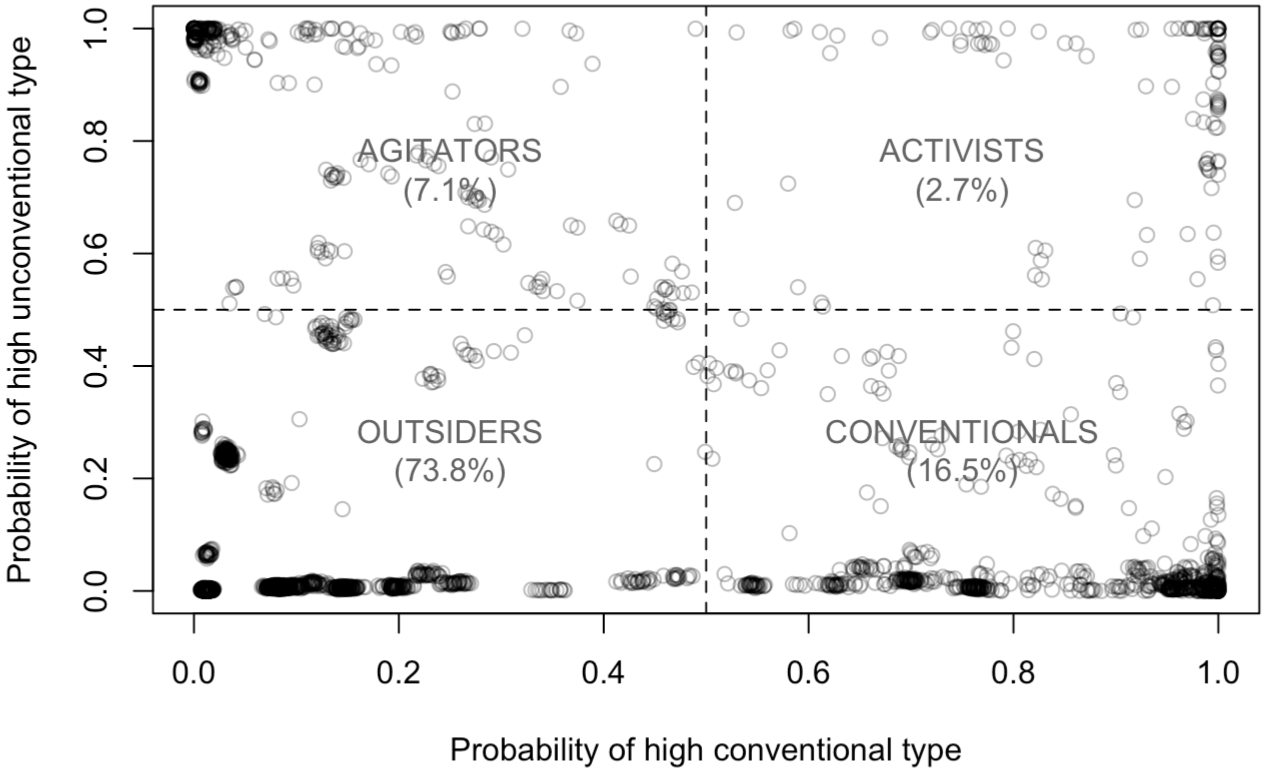
```r
pred.change.P[ , , 3] <- pred.P[ , , 4] - pred.P[ , , 1]

mean.prob <- apply(pred.P, c(2,3), mean)  * 100
rownames(mean.prob) <- act.names
quantile.prob.low <- apply(pred.P, c(2, 3), quantile, p = c(0.025))  * 100
quantile.prob.high <- apply(pred.P, c(2, 3), quantile, p = c(0.975))  * 100

mean.chprob <- apply(pred.change.P, c(2, 3), mean)  * 100
rownames(mean.chprob) <- act.names
quantile.chprob.low <- apply(pred.change.P, c(2, 3), quantile, p = c(0.025))  * 100
quantile.chprob.high <- apply(pred.change.P, c(2, 3), quantile, p = c(0.975))  * 100

T.probs <- cbind(mean.prob[, 1], quantile.prob.low[, 1], quantile.prob.high[, 1], mean.chp
colnames(T.probs) <- c("P(Y|T_LL)", "2.5%", "97.5%", "P(Y|T_LH) - P(Y|T_LL)", "2.5%", "97.
rownames(T.probs) <- act.names

conv.order <- order(T.ac.i , decreasing = TRUE)
```

```r
##-------- FIGURE 2 --------##

# The following bar plot gives participation probabilities for the four combined types:

#jpeg("Figure2.png", width = 750, height = 500)
barplot2(height = mean.prob[conv.order,], beside = TRUE, ci.l = quantile.prob.low[conv.ord
axis(2)
legend(0, 101, rownames(mean.prob[conv.order, ]), cex = 0.8, bty = "n", fill = rev(gray.co
mtext("Participation by Type",at = 24, line = 1, cex = 1)
```



```
#dev.off()

# For each combined type, activities are sorted based on the extent to which they are affe

## Type assignments

conv.type.samples <- rbind(samples.stage1[[1]][ , substr(colnames(samples.stage1[[1]]), 1,

unconv.type.samples <- rbind(samples.stage1[[1]][ , substr(colnames(samples.stage1[[1]]),

# Participatory types are not fixed for each individual; they are determined probabilistic

prob.conv.type <- apply(conv.type.samples - 1, 2, mean)
prob.unconv.type <- apply(unconv.type.samples - 1, 2, mean)

per.outsiders <- paste(round(mean(ifelse(prob.conv.type < 0.5 & prob.unconv.type < 0.5, 1,
per.engaged <- paste(round(mean(ifelse(prob.conv.type >= 0.5 & prob.unconv.type < 0.5, 1,
per.agitators <- paste(round(mean(ifelse(prob.conv.type < 0.5 & prob.unconv.type >= 0.5, 1
per.factotums <- paste(round(mean(ifelse(prob.conv.type >= 0.5 & prob.unconv.type >= 0.5,
```

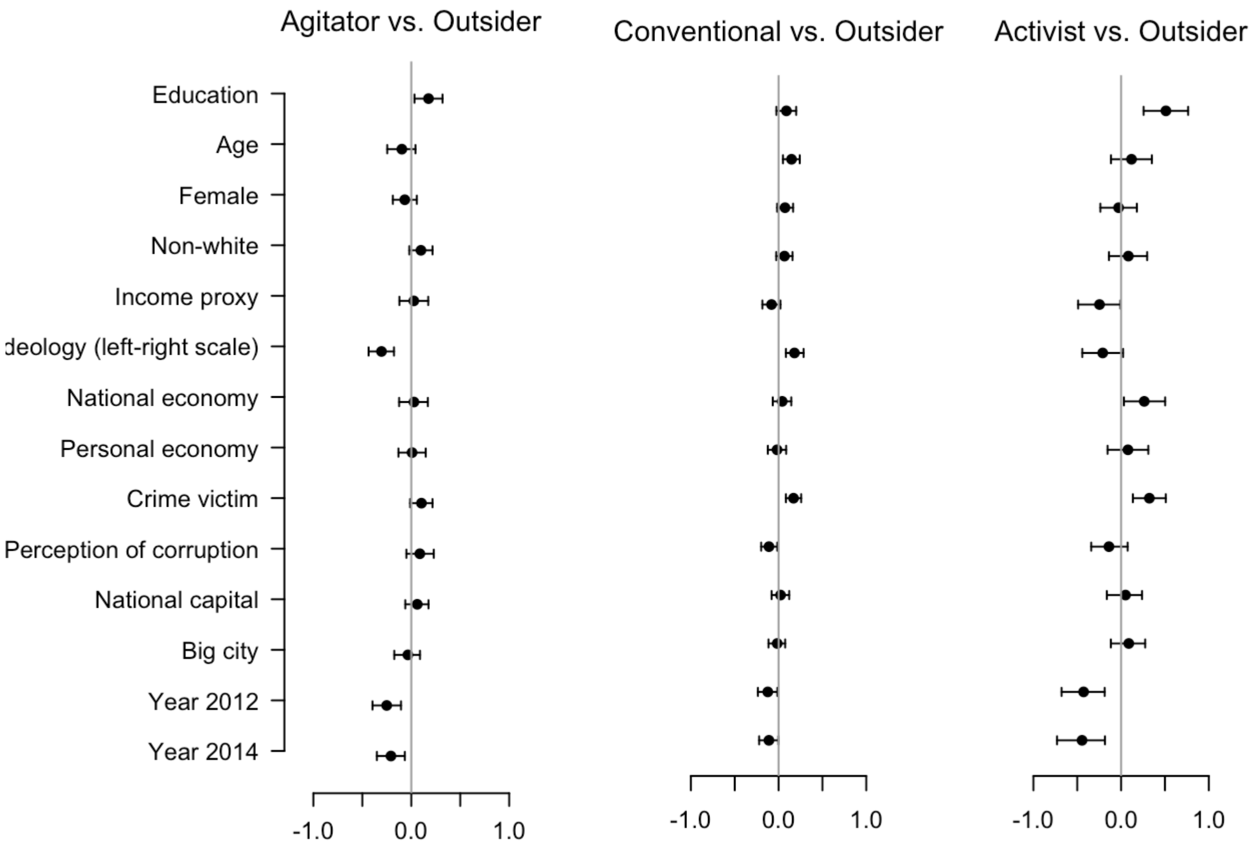
```
##-------- FIGURE 1 --------##

# The following scatterplot gives the relationship between the probability of being assign

#jpeg("Figure1.png", width = 600, height = 600)
plot(prob.conv.type, prob.unconv.type, xlab = "Probability of high conventional type",ylab
abline(h = 0.5, lty = 2)
abline(v = 0.5, lty = 2)
text(0.25, 0.25, paste("OUTSIDERS\n", "(",per.outsiders,")", sep = ""), col = "grey40")
text(0.75, 0.25, paste("CONVENTIONALS\n", "(",per.engaged,")", sep = ""), col = "grey40")
text(0.25, 0.75, paste("AGITATORS\n", "(",per.agitators,")", sep = ""), col = "grey40")
text(0.75, 0.75, paste("ACTIVISTS\n", "(",per.factotums,")", sep = ""), col = "grey40")
```




```
#dev.off()
```

```
# Goal: Make Figure 3
# Dependencies: "samples_stage2.Rdata"

library(coda)
library(plotrix)

load("samples_stage2.Rdata")

samples.stage2 <- as.mcmc.list(samples.stage2)

coef.samples <- samples.stage2[, substr(names(samples.stage2[[1]][1,]), 1, 6) != "beta[1"]

coef.samples.all30 <- NULL
for (j in 1:30) {
  coef.samples.all30 <- rbind(coef.samples.all30, coef.samples[[j]])
}

N.iters <- dim(coef.samples.all30)[1]

beta.samples.type2 <- coef.samples.all30[, substr(colnames(coef.samples.all30), 1, 6) == "
beta.samples.type3 <- coef.samples.all30[, substr(colnames(coef.samples.all30), 1, 6) == "
beta.samples.type4 <- coef.samples.all30[, substr(colnames(coef.samples.all30), 1, 6) == "

beta.means.type2 <- apply(beta.samples.type2, 2, mean)
beta.means.type3 <- apply(beta.samples.type3, 2, mean)
beta.means.type4 <- apply(beta.samples.type4, 2, mean)
```

```
beta.ql.type2 <- apply(beta.samples.type2, 2, quantile, p = 0.025)
beta.ql.type3 <- apply(beta.samples.type3, 2, quantile, p = 0.025)
beta.ql.type4 <- apply(beta.samples.type4, 2, quantile, p = 0.025)

beta.qu.type2 <- apply(beta.samples.type2, 2, quantile, p = 0.975)
beta.qu.type3 <- apply(beta.samples.type3, 2, quantile, p = 0.975)
beta.qu.type4 <- apply(beta.samples.type4, 2, quantile, p = 0.975)

##-------- FIGURE 3 --------##

varnames <- c("Education", "Age", "Female", "Non-white", "Income proxy", "Ideology (left-r

#jpeg("Figure3.png", width = 850, height = 500)

par(mfrow = c(1, 3))
par(oma = c(5, 11, 0, 0), mar = c(0, 0, 4, 4))

# (L, H) vs. (L, L)

plotCI(y = 1:14 - 0.1, x = rev(beta.means.type2[2:15]), err = "x", ui = rev(beta.qu.type2[
axis(side = 1 , cex.axis = 1)
axis(side = 2 , las = 1, at = c(1:14), cex.axis = 1,labels = rev(varnames), las = 2)
mtext("Agitator vs. Outsider", at = 0, line = 1, cex = 0.9)
abline(v = 0, col = "grey63")

# (H, L) vs. (L, L)

plotCI(y = 1:14 - 0.1, x = rev(beta.means.type3[2:15]), err = "x", ui = rev(beta.qu.type3[
axis(side = 1, cex.axis = 1)
mtext("Conventional vs. Outsider",at = 0, line = 1, cex = 0.9)
abline(v=0,col="grey63")

# (H, H) vs. (L, L)

plotCI(y = 1:14 - 0.1, x = rev(beta.means.type4[2:15]), err = "x", ui = rev(beta.qu.type4[
axis(side = 1, cex.axis = 1)
mtext("Activist vs. Outsider", at = 0, line = 1, cex = 0.9)
abline(v = 0, col = "grey63")
```




```
#dev.off()
```

```
# Goal: Make Figures 4-6 and Table C1
# Dependencies: "datamodel.Rdata", "samples_stage2.Rdata"

library(coda)
library(plotrix)
library(xtable)

# Load recoded data

load("datamodel.Rdata")

datamodel2 <- data.frame("education" = datamodel$education, "age" = datamodel$age, "female

# Standardize covariates
datamodel2 <- as.data.frame(scale(datamodel2))

# Load 2nd stage chains
load("samples_stage2.Rdata")

samples.stage2 <- as.mcmc.list(samples.stage2)

coef.samples <- samples.stage2[ , substr(names(samples.stage2[[1]][1, ]), 1, 6) != "beta[1

coef.samples.all <- NULL
for (j in 1:30) {
  coef.samples.all <- rbind(coef.samples.all, coef.samples[[j]])
```

```r
}

N.iters <- dim(coef.samples.all)[1]

n.sims <- 15

X.median <- c(1, apply(datamodel2, 2, median))
names(X.median)[1] <- "constant"
X.median["female"] <- quantile(datamodel2$female)[2]
X.median["natecon"] <- quantile(datamodel2$natecon)[5]
X.median["persfin"] <- quantile(datamodel2$persfin)[5]
X.median["corrupt"] <- quantile(datamodel2$corrupt, p = c(0.05))

X.sims <- matrix(rep(X.median), nrow = n.sims, ncol = length(X.median), byrow = T)
colnames(X.sims) <- names(X.median)

X.sims[2, "education"] <- quantile(datamodel2$education)[4]
X.sims[3, "age"] <- quantile(datamodel2$age)[4]
X.sims[4, "female"] <- quantile(datamodel2$female)[4]
X.sims[5, "nonwhite"] <- quantile(datamodel2$nonwhite)[4]
X.sims[6, "income.proxy"] <- quantile(datamodel2$income.proxy)[4]
X.sims[7, "ideology"] <- quantile(datamodel2$ideology)[4]
X.sims[8, "natecon"] <- quantile(datamodel2$natecon)[2]
X.sims[9, "persfin"] <- quantile(datamodel2$persfin)[2]
X.sims[10, "victim"] <- quantile(datamodel2$victim)[5]
X.sims[11, "corrupt"] <- quantile(datamodel2$corrupt)[4]
X.sims[12, "capital"] <- quantile(datamodel2$capital)[5]
X.sims[13, "bigcity"] <- quantile(datamodel2$bigcity)[5]
X.sims[14, "y2012"] <- quantile(datamodel2$y2012)[5]
X.sims[15, "y2014"] <- quantile(datamodel2$y2014)[5]

betas.list <- list()
eXB.list <- list()
P.list <- list()

for (j in 2:4) {

  betas.list[[j]] <- coef.samples.all[ , substr(colnames(coef.samples.all), 1, 6) == paste

  eXB.list[[j]] <- exp(betas.list[[j]] %*% t(X.sims))

}

P.list[[2]] <- eXB.list[[2]] / (matrix(1, nrow = N.iters, ncol = 15) + eXB.list[[2]] + eXB
P.list[[3]] <- eXB.list[[3]] / (matrix(1, nrow = N.iters, ncol = 15) + eXB.list[[2]] + eXB
P.list[[4]] <- eXB.list[[4]] / (matrix(1, nrow = N.iters, ncol = 15) + eXB.list[[2]] + eXB
P.list[[1]] <- matrix(1, nrow = N.iters, ncol = 15) - P.list[[2]] - P.list[[3]] - P.list[[

Base.mean.list <- list()
Base.ql.list <- list()
Base.qu.list <- list()
dP.list <- list()
dP.mean.list <- list()
dP.ql.list <- list()
dP.qu.list <- list()
```

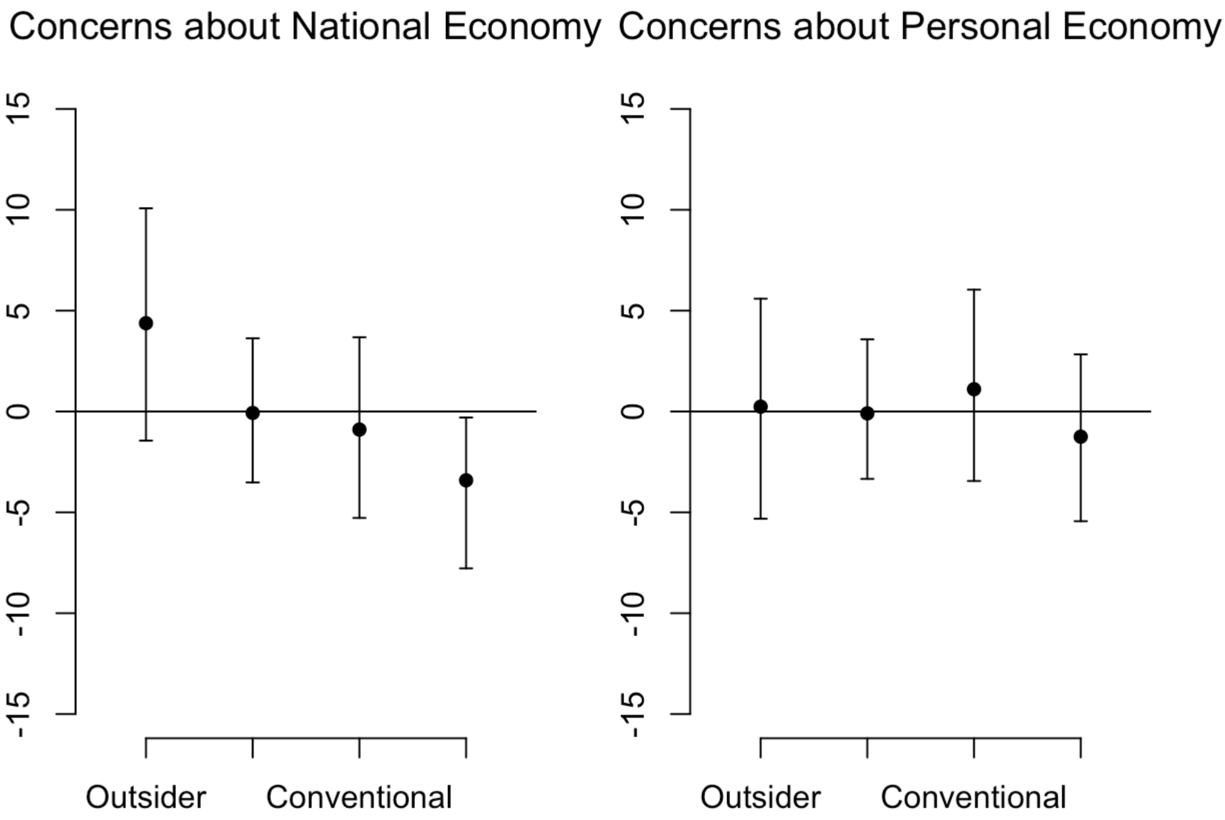
```
for (j in 1:4) {

  Base.mean.list[[j]] <- apply(P.list[[j]], 2, mean)

  Base.ql.list[[j]] <- apply(P.list[[j]], 2, quantile, p = 0.025)

  Base.qu.list[[j]] <- apply(P.list[[j]], 2, quantile, p = 0.975)

  dP.list[[j]] <- matrix(NA, nrow = N.iters, ncol = 14)

  for (k in 1:14) {
    dP.list[[j]][ , k] <- P.list[[j]][ , k + 1] - P.list[[j]][ , 1]
  }

  dP.mean.list[[j]] <- apply(dP.list[[j]], 2, mean)
  dP.ql.list[[j]] <- apply(dP.list[[j]], 2, quantile, p = 0.025)
  dP.qu.list[[j]] <- apply(dP.list[[j]], 2, quantile, p = 0.975)

  names(dP.mean.list[[j]]) <- names(X.median)[-1]
  names(dP.ql.list[[j]]) <- names(X.median)[-1]
  names(dP.qu.list[[j]]) <- names(X.median)[-1]

}
```

```
##-------- TABLE C1 --------##

METABLE <- cbind(
  dP.mean.list[[1]], dP.ql.list[[1]], dP.qu.list[[1]],
  dP.mean.list[[2]], dP.ql.list[[2]], dP.qu.list[[2]],
  dP.mean.list[[3]], dP.ql.list[[3]], dP.qu.list[[3]],
  dP.mean.list[[4]], dP.ql.list[[4]], dP.qu.list[[4]])

TableC1 <- rbind(c(Base.mean.list[[1]][1], Base.ql.list[[1]][1], Base.qu.list[[1]][1], Bas

colnames(TableC1) <- c("P(Outsider)", "2.5%", "97.5%", "P(Agitator)", "2.5%", "97.5%", "P(

# for latex
xtable(TableC1, digits  =1)
```

```
% latex table generated in R 4.4.2 by xtable 1.8-4 package
% Sat May 31 01:44:14 2025
\begin{table}[ht]
\centering
\begin{tabular}{rrrrrrrrrrrrr}
  \hline
 & P(Outsider) & 2.5\% & 97.5\% & P(Agitator) & 2.5\% & 97.5\% & P(Conventional) & 2.5\% &
97.5\% & P(Activist) & 2.5\% & 97.5\% \\
  \hline
 & 64.5 & 58.0 & 70.9 & 9.5 & 6.1 & 13.7 & 19.0 & 14.3 & 24.4 & 7.0 & 3.2 & 12.5 \\
  education & -2.4 & -3.8 & -1.1 & 0.5 & -0.1 & 1.3 & 0.1 & -0.8 & 1.1 & 1.7 & 0.6 & 3.5
\\
  age & -1.9 & -3.9 & 0.2 & -1.1 & -2.3 & 0.0 & 2.3 & 0.7 & 4.0 & 0.7 & -0.8 & 2.6 \\
  female & -0.9 & -4.5 & 2.6 & -1.3 & -3.4 & 0.7 & 2.7 & -0.3 & 5.8 & -0.5 & -3.3 & 2.3 \\
  nonwhite & -3.9 & -7.9 & 0.1 & 1.5 & -0.8 & 4.0 & 1.6 & -1.6 & 4.7 & 0.9 & -2.4 & 4.4 \\
  income.proxy & 1.3 & -0.1 & 2.8 & 0.4 & -0.5 & 1.4 & -0.7 & -1.9 & 0.4 & -1.0 & -2.3 &
```

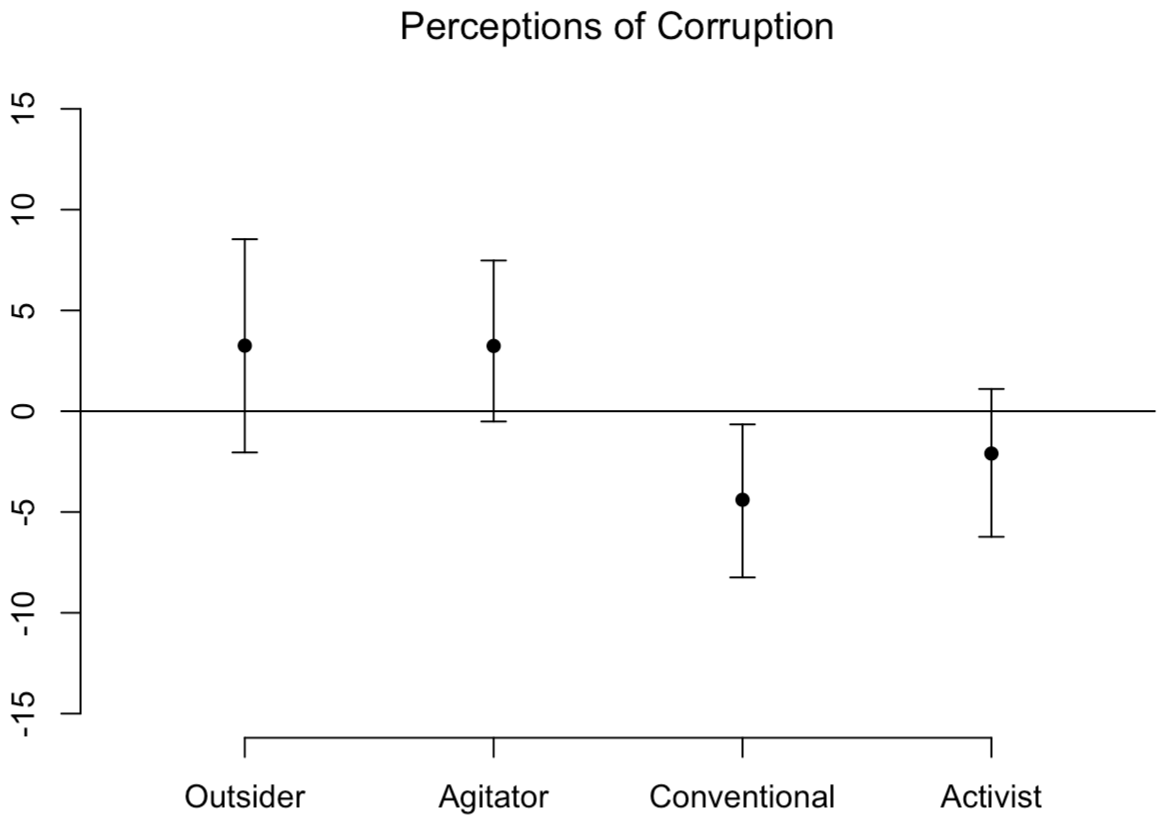
```
0.0 \\
  ideology & 0.0 & -2.2 & 2.2 & -2.4 & -3.8 & -1.4 & 3.7 & 2.1 & 5.6 & -1.3 & -3.0 & 0.0
\\
  natecon & 4.4 & -1.4 & 10.1 & -0.1 & -3.5 & 3.6 & -0.9 & -5.3 & 3.7 & -3.4 & -7.8 & -0.3
\\
  persfin & 0.2 & -5.3 & 5.6 & -0.1 & -3.3 & 3.6 & 1.1 & -3.4 & 6.0 & -1.2 & -5.4 & 2.8 \\
  victim & -10.7 & -15.6 & -6.2 & 0.7 & -1.7 & 3.3 & 4.6 & 0.9 & 8.5 & 5.4 & 1.3 & 11.7 \\
  corrupt & 3.3 & -2.0 & 8.5 & 3.2 & -0.5 & 7.5 & -4.4 & -8.2 & -0.7 & -2.1 & -6.2 & 1.1
\\
  capital & -3.6 & -10.4 & 3.1 & 1.8 & -2.2 & 6.4 & 0.6 & -4.7 & 6.7 & 1.2 & -3.7 & 7.0 \\
  bigcity & -0.3 & -6.0 & 5.0 & -0.9 & -3.9 & 2.2 & -0.9 & -4.9 & 3.2 & 2.1 & -2.0 & 7.3
\\
  y2012 & 9.1 & 4.9 & 13.5 & -3.1 & -5.7 & -0.8 & -2.3 & -6.0 & 1.1 & -3.7 & -7.8 & -1.1
\\
  y2014 & 8.3 & 4.1 & 12.7 & -2.5 & -5.0 & -0.3 & -2.0 & -5.5 & 1.5 & -3.8 & -7.7 & -1.2
\\
   \hline
\end{tabular}
\end{table}
```

```
##-------- FIGURE 4 --------##

# Effect economic evaluations

#jpeg("Figure4.png", width = 800, height = 400)

par(mfrow=c(1, 2))
par(oma=c(4, 1, 4, 2),mar=c(0, 4, 0, 0))

plotCI(x = 1:4, y = TableC1["natecon", c(1, 4, 7, 10)], li = TableC1["natecon", c(2, 5, 8,
axis(2, cex.axis = 1)
axis(1, at = 1:4, labels = c("Outsider", "Agitator", "Conventional", "Activist"), cex.axis
mtext("Concerns about National Economy",at = 2.5, line = 1, cex = 1.2)
abline(h = 0, col = "black")

plotCI(x = 1:4, y = TableC1["persfin", c(1, 4, 7, 10)], li = TableC1["persfin", c(2, 5, 8,
axis(2, cex.axis = 1)
axis(1, at = 1:4, labels = c("Outsider", "Agitator", "Conventional", "Activist"), cex.axis
mtext("Concerns about Personal Economy",at = 2.5, line = 1, cex = 1.2)
abline(h = 0, col = "black")

mtext("Economic Evaluations", at = 0.46 , side = 3, line = -40, cex = 1, font = 2, outer =
```



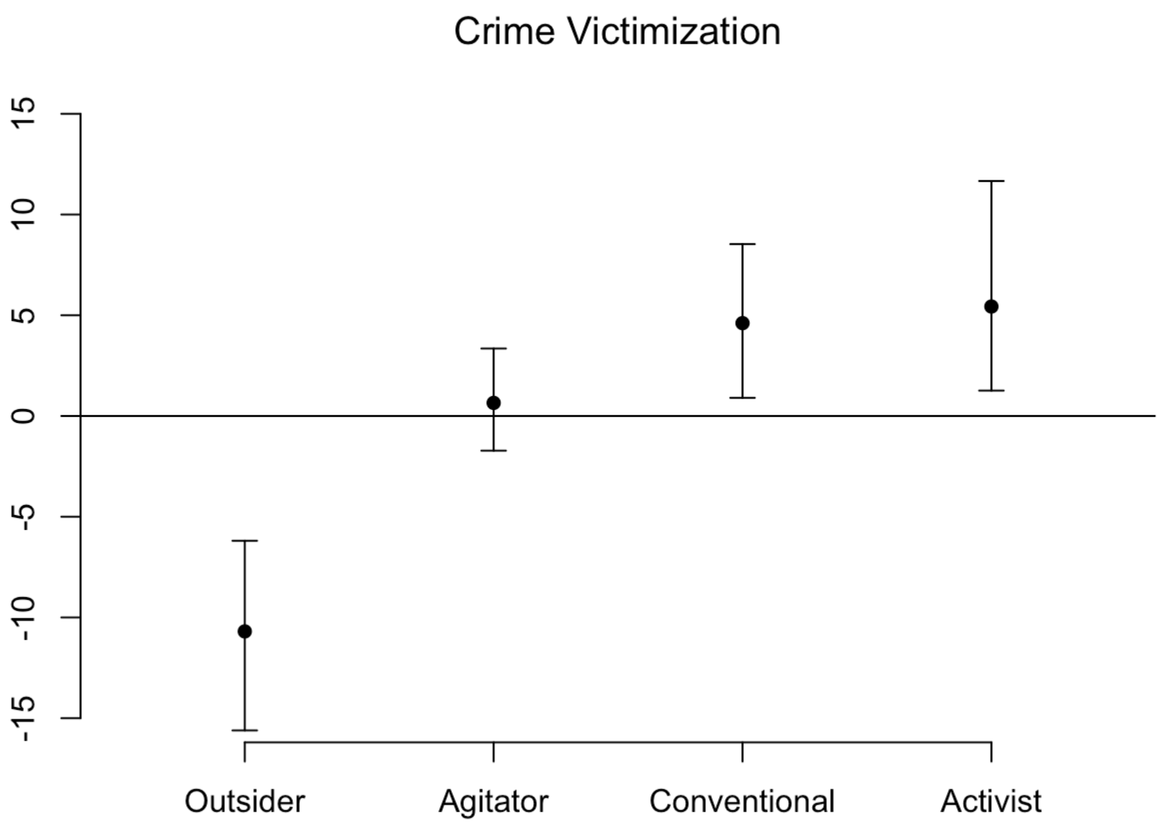

```
#dev.off()
```

```
##-------- FIGURE 5 --------##

# Effect perceptions of corruption

#jpeg("Figure5.png", width = 500, height = 400)

par(mfrow=c(1, 1))
par(oma=c(4, 1, 4, 2),mar=c(0, 4, 0, 0))

plotCI(x = 1:4, y = TableC1["corrupt", c(1, 4, 7, 10)], li = TableC1["corrupt", c(2, 5, 8,
axis(2, cex.axis = 1)
axis(1, at = 1:4, labels = c("Outsider", "Agitator", "Conventional", "Activist"), cex.axis
mtext("Perceptions of Corruption", at = 2.5, line = 1, cex = 1.2)
abline(h = 0, col = "black")
```




```
#dev.off()
```

```
##-------- FIGURE 6 --------##

# Effect crime victimization

#jpeg("Figure6.png", width = 500, height = 400)

par(mfrow=c(1, 1))
par(oma=c(4, 1, 4, 2),mar=c(0, 4, 0, 0))

plotCI(x = 1:4, y = TableC1["victim", c(1, 4, 7, 10)], li = TableC1["victim", c(2, 5, 8, 1
axis(2, cex.axis = 1)
axis(1, at = 1:4, labels = c("Outsider", "Agitator", "Conventional", "Activist"), cex.axis
mtext("Crime Victimization", at = 2.5, line = 1, cex = 1.2)
abline(h = 0, col = "black")
```




```
#dev.off()
```

Comparison

```
# DIC
# model.jags saved to save time

#library(rjags)
#library(coda)
#load.module("glm")

#library(foreach)
#library(doMC)
#registerDoMC(4)

#RNGkind("L'Ecuyer-CMRG")
#set.seed(100)
# Load recoded data

#load("datamodel.Rdata")

#attach(datamodel)
#n <- dim(datamodel)[1]
#fixed.unconv <- rep(0, n)
#fixed.unconv[unconv.scale == 3 & conv.scale == 0] <- 2
#fixed.unconv[unconv.scale == 0 & conv.scale > 6] <- 1
#which(fixed.unconv != 0)
```

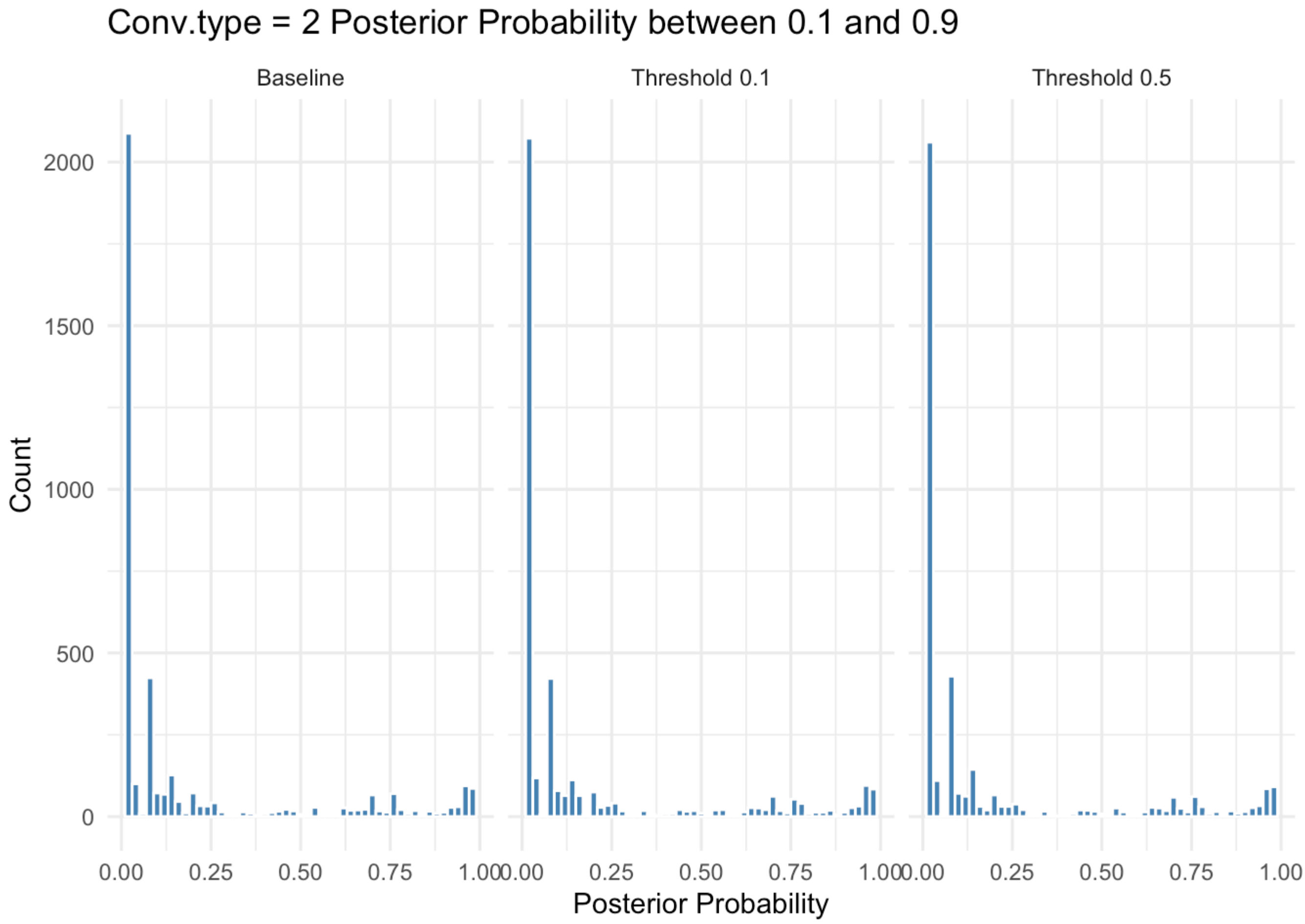
```
#fixed.conv <- rep(0, n)
#fixed.conv[unconv.scale == 3 & conv.scale == 0] <- 1
#fixed.conv[unconv.scale == 0 & conv.scale > 6] <- 2
#which(fixed.conv != 0)

#Y <- cbind(act.meet.mun, act.cont.mun, act.cont.aut, act.meet.imp, act.meet.pty, act.meet
#ntypes <- 2
#nact <- dim(Y)[2]
#nconv <- nact - 3
#wprior <- c(1, 1)

#data.jags  <- list("n" = n, "wprior" = wprior, "ntypes" = ntypes, "nact" = nact, "Y"=  as

#inits.jags <- list(
#  list(.RNG.seed = sample(1:10000, 1), .RNG.name = "base::Mersenne-Twister"),
#  list(.RNG.seed = sample(1:10000, 1), .RNG.name = "base::Mersenne-Twister")
#)
#model.jags <- jags.model("stage1.bug", data = data.jags, inits = inits.jags, n.chains = 2
#save(model.jags, file = "model_jags.Rdata")

#dic_result <- dic.samples(model.jags, n.iter = 1000, type = "pD")
#dic_value <- sum(dic_result$deviance + dic_result$penalty)
#print(dic_value)

#load("model_jags_0.1.Rdata")
#dic_result_0.1 <- dic.samples(model.jags_0.1, n.iter = 1000, type = "pD")
#dic_value_0.1 <- sum(dic_result_0.1$deviance + dic_result_0.1$penalty)
#print(dic_value_0.1)
```

```
# visualization for comparison
library(coda)
library(ggplot2)
library(dplyr)

files <- c("samples_stage1.Rdata", "samples_stage1_0.1.Rdata", "samples_stage1_0.5.Rdata")
labels <- c("Baseline", "Threshold 0.1", "Threshold 0.5")

results <- data.frame()
plot_data <- data.frame()

for (i in seq_along(files)) {

  load(files[i])
  combined.samples <- as.mcmc(do.call(rbind, samples.stage1))
  n <- length(grep("^conv\\.type\\[", colnames(combined.samples)))

  conv.probs <- numeric(n)
  unconv.probs <- numeric(n)

  for (j in 1:n) {
    conv.col <- paste0("conv.type[", j, "]")
    unconv.col <- paste0("unconv.type[", j, "]")

    conv.probs[j] <- mean(combined.samples[, conv.col] == 2)
```

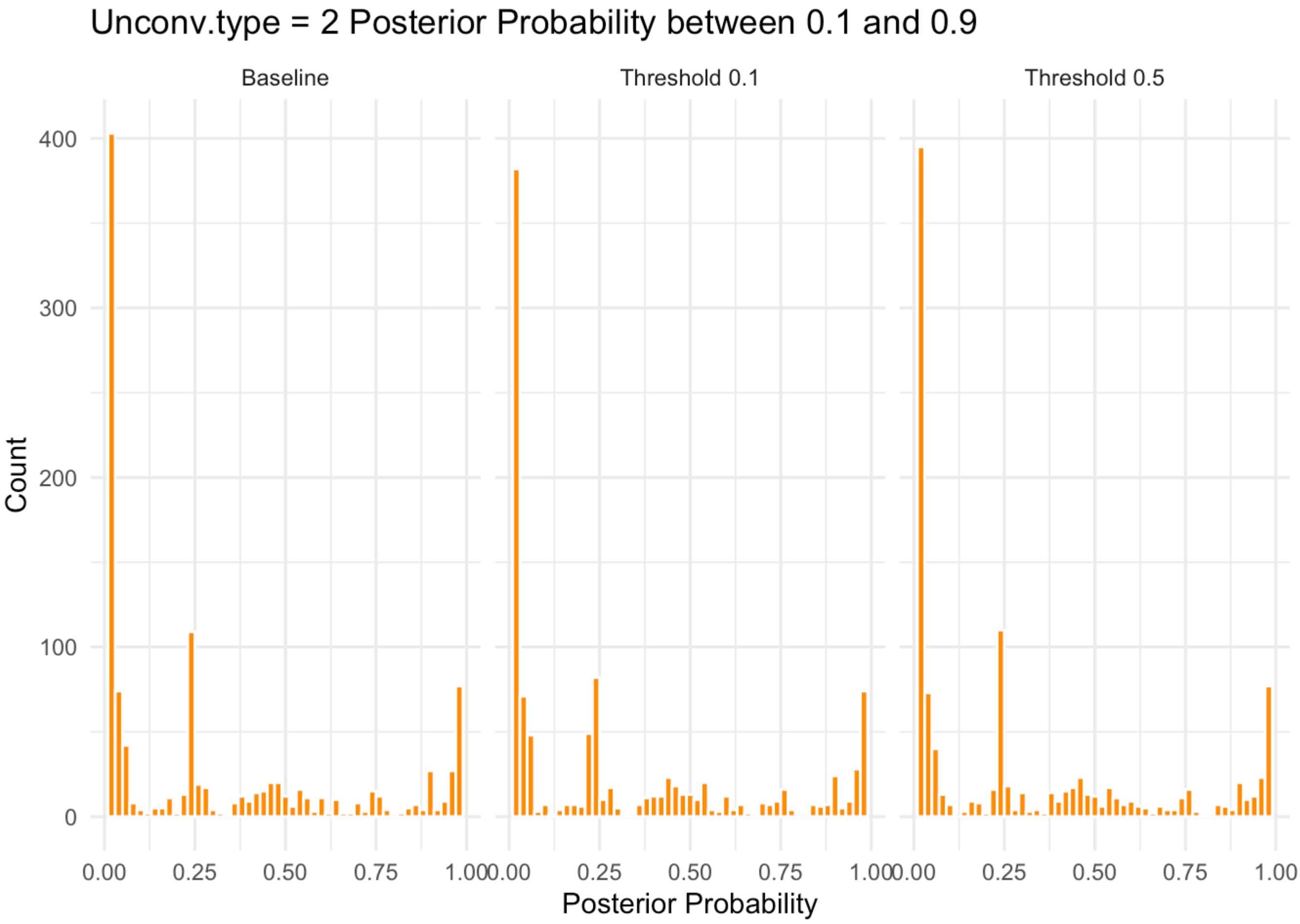
```r
    unconv.probs[j] <- mean(combined.samples[, unconv.col] == 2)
  }

  p.conv.uncertain <- mean(conv.probs > 0.45 & conv.probs < 0.55)
  p.unconv.uncertain <- mean(unconv.probs > 0.45 & unconv.probs < 0.55)

  results <- rbind(results, data.frame(
    Setting = labels[i],
    Uncertain_Conv = round(p.conv.uncertain, 3),
    Uncertain_Unconv = round(p.unconv.uncertain, 3)
  ))

  df_tmp <- data.frame(
    conv_prob = conv.probs,
    unconv_prob = unconv.probs,
    Setting = labels[i]
  )
  df_tmp <- df_tmp %>%
    filter(conv_prob > 0 | unconv_prob > 0)

  plot_data <- rbind(plot_data, df_tmp)
}

print(results)
```

```
        Setting Uncertain_Conv Uncertain_Unconv
1      Baseline          0.018            0.017
2 Threshold 0.1          0.015            0.017
3 Threshold 0.5          0.017            0.016
```

```r
ggplot(plot_data %>% filter(conv_prob >= 0.01 & conv_prob <= 0.99), aes(x = conv_prob)) +
  geom_histogram(binwidth = 0.02, fill = "steelblue", color = "white") +
  facet_wrap(~Setting) +
  labs(title = "Conv.type = 2 Posterior Probability between 0.1 and 0.9",
       x = "Posterior Probability", y = "Count") +
  theme_minimal()
```




```
ggplot(plot_data %>% filter(unconv_prob >= 0.01 & unconv_prob <= 0.99), aes(x = unconv_pro
  geom_histogram(binwidth = 0.02, fill = "darkorange", color = "white") +
  facet_wrap(~Setting) +
  labs(title = "Unconv.type = 2 Posterior Probability between 0.1 and 0.9",
       x = "Posterior Probability", y = "Count") +
  theme_minimal()
```



Unconv.type = 2 Posterior Probability between 0.1 and 0.9
Baseline
Threshold 0.1
Threshold 0.5
Count
Posterior Probability
400
300
200
100
0
0.00
0.25
0.50
0.75
1.00