Predicted probabilities after Bayesian logit/probit models

Johannes Karreth

July 5, 2016

Purpose

This document shows how to use two functions I wrote, `MCMC_simcase_probs` and `MCMC_observed_probs`, to quickly calculate predicted probabilities for “average” and “observed” cases based on estimates from Bayesian logit or probit models (for binary outcomes). For more details on each approach, see King et al. (AJPS 2000) and Hanmer & Kalkan (AJPS 2013), listed in the syllabus.

Both functions can be accessed from my Github repository at https://github.com/jkarreth/JKmisc. **These functions are not fully tested and should be used with extreme caution. Please let me know right away about any problems you run into so I can improve these functions.**

The functions require the following R packages to be installed: dplyr and reshape2.

Example 1

Source: Hainmueller, J. and Hiscox, M. J. (2006). Learning to Love Globalization: Education and Individual Attitudes Toward International Trade. International Organization, 60 (2):469-498.

This example analyzes a simplified version of one of the empirical models in Hainmueller and Hiscox’ study of individuals’ support for free trade or protectionist policies. The data are a cleaned up and modified version of the 1996 American National Election studies as used in Hainmueller & Hiscox (2006). The outcome variable `protectionist` is binary - coded as 1 if a respondent expressed a preference for more protectionist policies, and coded as 0 if a respondent favored free trade. The explanatory variables are (see Table A2 in Hainmueller & Hiscox (2006) for more details):

- `age`: the respondent’s age in years.
- `female`: a binary indicator for female respondents.
- `TUmember`: a binary indicator for trade union members.
- `partyid`: the respondent’s party identification: coded from 0 “strong Democrat”" to 6 “strong Republican”.
- `ideology`: the respondent’s ideology: coded 0 if conservative, 1 if moderate, and 2 if liberal.
- `schooling`: years of full-time education completed.

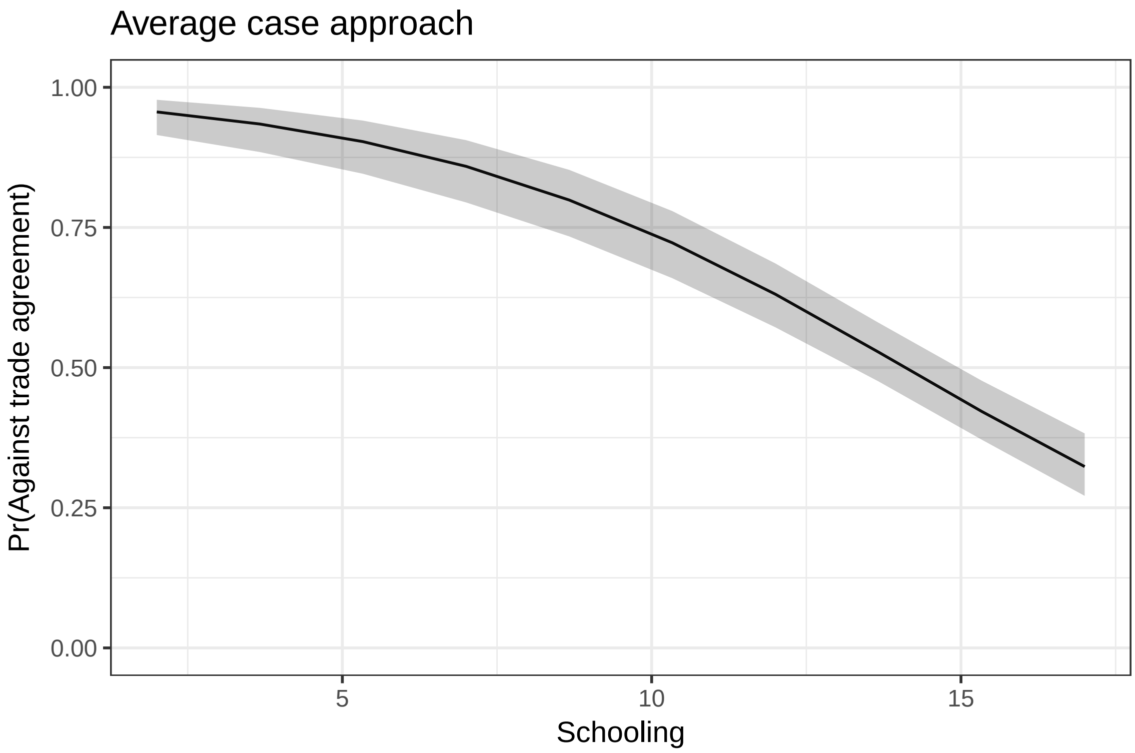
The estimated model is a Bayesian logistic regression model, fit in JAGS.

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 817
##    Unobserved stochastic nodes: 7
##    Total graph size: 7394
##
## Initializing model
##              Mean    SD    Lower    Upper    Pr
## b[1]        3.645 0.588    2.429    4.792 1.000
## b[2]        0.005 0.005   -0.004    0.015 0.876
## b[3]        0.615 0.156    0.322    0.935 1.000
## b[4]        0.221 0.225   -0.232    0.644 0.837
## b[5]       -0.100 0.042   -0.183   -0.021 0.994
## b[6]       -0.288 0.103   -0.487   -0.081 0.997
## b[7]       -0.255 0.033   -0.319   -0.188 1.000
## deviance 1022.585 3.942 1017.320 1031.976 1.000
```

Probabilities

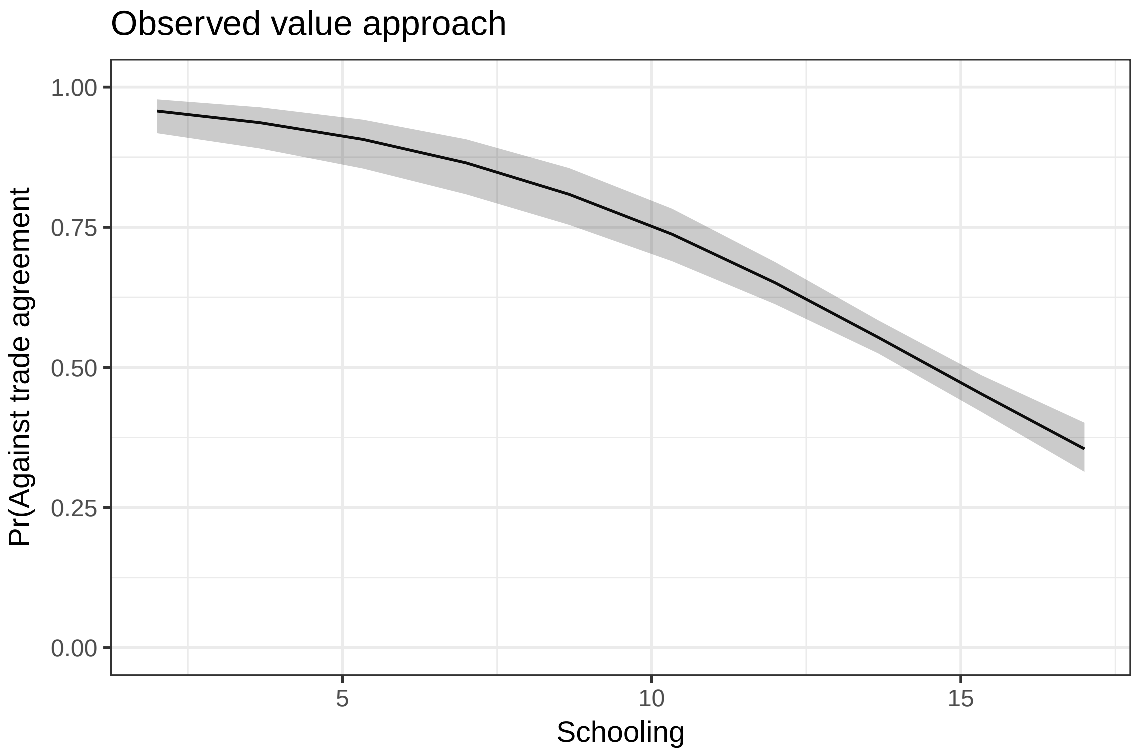
Average case approach

```
devtools::source_url("https://raw.githubusercontent.com/jkarreth/JKmisc/master/MCMC_simcase_probs.R")

prot_xmat <- model.matrix(protectionist ~ age + female + TUmember + partyid +
                            ideology + schooling,
                          data = prot.dat)

prot_mcmc <- as.mcmc(prot.fit)
prot_mcmc_mat <- as.matrix(prot_mcmc)[, 1:ncol(prot_xmat)]

prot_schooling_sim <- seq(from = min(prot.dat$schooling),
                          to = max(prot.dat$schooling),
                          length.out = 10)

prot_sim_prob <- MCMC_simcase_probs(model_matrix = prot_xmat,
                mcmc_out = prot_mcmc_mat,
                x_col = 7,
                x_range_vec = prot_schooling_sim)

library(ggplot2)
p_sim <- ggplot(data = prot_sim_prob, aes(x = predictor, y = median_pp)) +
  geom_line() + geom_ribbon(aes(ymin = lower_pp, ymax = upper_pp), alpha = 0.25) +
  ylim(c(0, 1)) + xlab("Schooling") + ylab("Pr(Against trade agreement)") +
  ggtitle("Average case approach") + theme_bw()
p_sim
```



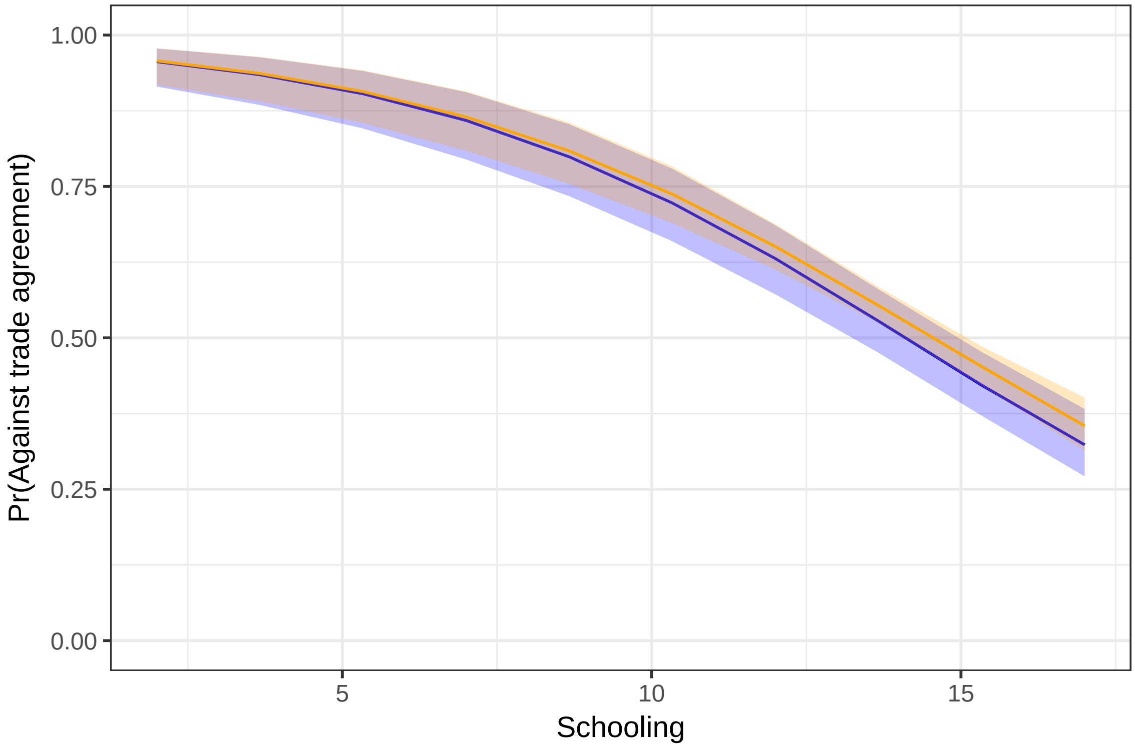
Observed value approach

```r
devtools::source_url("https://raw.githubusercontent.com/jkarreth/JKmisc/master/MCMC_observed_probs.R")

prot_xmat <- model.matrix(protectionist ~ age + female + TUmember + partyid +
                            ideology + schooling,
                          data = prot.dat)

prot_mcmc <- as.mcmc(prot.fit)
prot_mcmc_mat <- as.matrix(prot_mcmc)[, 1:ncol(prot_xmat)]

prot_schooling_sim <- seq(from = min(prot.dat$schooling),
                          to = max(prot.dat$schooling),
                          length.out = 10)

prot_obs_prob <- MCMC_observed_probs(model_matrix = prot_xmat,
                mcmc_out = prot_mcmc_mat,
                x_col = 7,
                x_range_vec = prot_schooling_sim)

library(ggplot2)
p_obs <- ggplot(data = prot_obs_prob, aes(x = predictor, y = median_pp)) +
  geom_line() + geom_ribbon(aes(ymin = lower_pp, ymax = upper_pp), alpha = 0.25) +
  ylim(c(0, 1)) + xlab("Schooling") + ylab("Pr(Against trade agreement") +
  ggtitle("Observed value approach") + theme_bw()
p_obs
```



Comparing both approaches

```
p_combined <- ggplot(data = prot_sim_prob, aes(x = predictor, y = median_pp)) +
  geom_line(color = "blue") +
  geom_ribbon(aes(ymin = lower_pp, ymax = upper_pp), fill = "blue", alpha = 0.25) +
  ylim(c(0, 1)) +
  geom_line(data = prot_obs_prob, aes(x = predictor, y = median_pp), color = "orange") +
  geom_ribbon(data = prot_obs_prob, aes(ymin = lower_pp, ymax = upper_pp), fill = "orange", alpha = 0.25
  xlab("Schooling") + ylab("Pr(Against trade agreement)") + theme_bw()

p_combined
```



Example 2

Source: Epstein, L., Lindstädt, R., Segal, J. A., and Westerland, C. (2006). The Changing Dynamics of Senate Voting on Supreme Court Nominees. Journal of Politics, 68 (2): 296–307.

This example is a simplified version of the "Additional nominees" model in Table 2 of Epstein et al. (2006). Epstein et al. examine why U.S. senators cast votes in favor or against nominees for the U.S. Supreme Court. The data contain 3709 observations, with each observation being one senator's vote, from 1937 to 2005. The outcome variable `vote` is binary - coded as 1 if a senator voted Yea on a candidate, and coded as 0 if the senator voted against the nominee. The explanatory variables are (see p. 298 in Epstein et al. (2006) for more details):

- `lackqual`: the degree to which senators perceive the candidate as qualified for office.
- `eucldist`: the ideological distance between the senator and the candidate.
- `strngprs`: a binary indicator for whether the president was "strong"" in the sense that his party controlled the Senate and he was not in his fourth year of office.
- `sameprty`: a binary indicator for whether a senator is of the same political party as the president.

The estimated model is a Bayesian logistic regression model, fit in JAGS.

```
nom.dat <- rio::import("http://epstein.wustl.edu/research/Bork.dta")

nom.datjags <- as.list(na.omit(nom.dat[, c("vote", "lackqual", "eucldist",
                                           "strngprs", "sameprty")]))
nom.datjags$N <- length(nom.datjags$vote)

nom.mod <- function()  {

  for(i in 1:N){
    vote[i] ~ dbern(p[i])  ## Bernoulli distribution of y_i
    logit(p[i]) <- mu[i]    ## Logit link function
    mu[i] <- b[1] + b[2] * lackqual[i] + b[3] * eucldist[i] + b[4] * strngprs[i]
    + b[5] * sameprty[i]
  }

  for(j in 1:5){
    b[j] ~ dnorm(0, 0.01) ## Use a coefficient vector for simplicity
  }

}

nom.params <- c("b")
nom.inits1 <- list("b" = rep(0, 5))
nom.inits2 <- list("b" = rep(0, 5))
nom.inits <- list(nom.inits1, nom.inits2)

set.seed(123)

library(R2jags)
nom.fit <- jags(data = nom.datjags, inits = nom.inits,
                parameters.to.save = nom.params, n.chains = 2, n.iter = 10000,
                n.burnin = 5000, n.thin = 5, model.file = nom.mod)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 3709
##    Unobserved stochastic nodes: 5
##    Total graph size: 28439
##
## Initializing model
```

```
devtools::source_url("https://raw.githubusercontent.com/jkarreth/JKmisc/master/mcmctab.R")
mcmctab(as.mcmc(nom.fit))
```

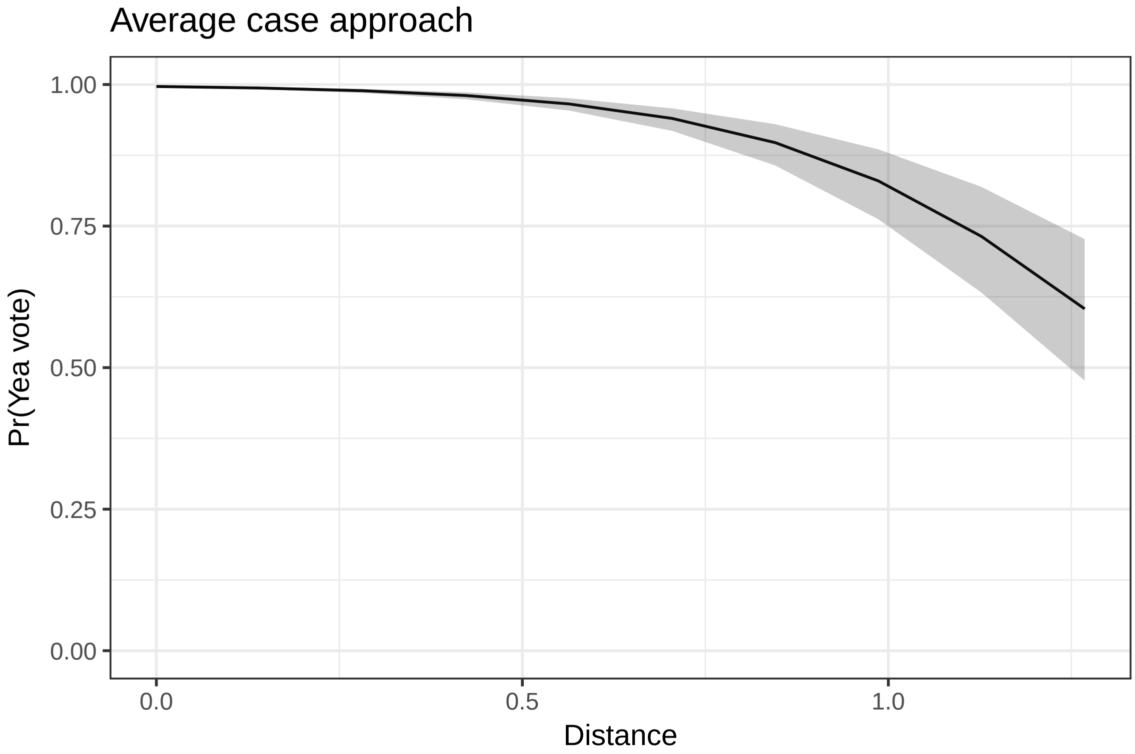
```
##              Mean     SD    Lower    Upper Pr
## b[1]        3.288  0.186    2.965    3.627  1
## b[2]       -4.367  0.272   -4.857   -3.908  1
## b[3]       -4.126  0.314   -4.705   -3.560  1
## b[4]        1.478  0.146    1.213    1.756  1
## b[5]        1.404  0.167    1.110    1.723  1
## deviance 1691.162 40.777 1685.256 1698.247  1
```

Probabilities

Average case approach

```
devtools::source_url("https://raw.githubusercontent.com/jkarreth/JKmisc/master/MCMC_simcase_probs.R")

nom_xmat <- model.matrix(vote ~ lackqual + eucldist + strngprs + sameprty,
                         data = nom.dat)
```

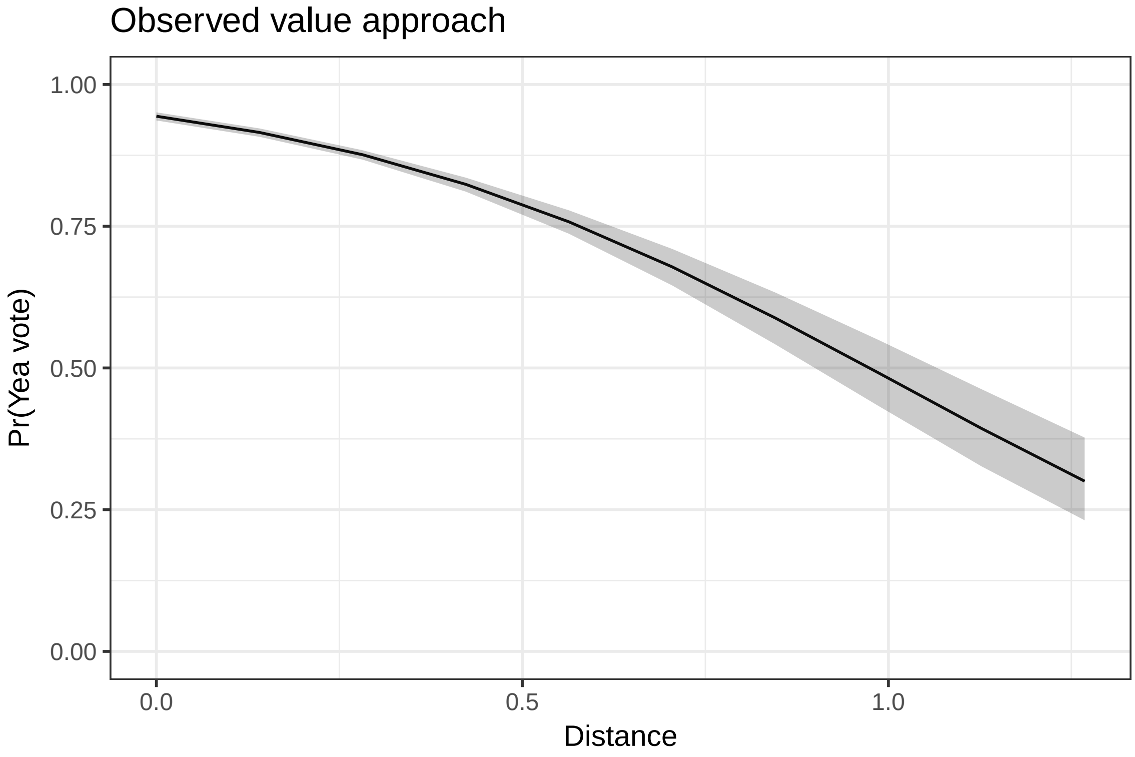
```
nom_mcmc <- as.mcmc(nom.fit)
nom_mcmc_mat <- as.matrix(nom_mcmc)[, 1:ncol(nom_xmat)]

nom_distance_sim <- seq(from = min(nom.dat$eucldist),
                        to = max(nom.dat$eucldist),
                        length.out = 10)

nom_sim_prob <- MCMC_simcase_probs(model_matrix = nom_xmat,
                mcmc_out = nom_mcmc_mat,
                x_col = 3,
                x_range_vec = nom_distance_sim)

library(ggplot2)
p_sim <- ggplot(data = nom_sim_prob, aes(x = predictor, y = median_pp)) +
  geom_line() + geom_ribbon(aes(ymin = lower_pp, ymax = upper_pp), alpha = 0.25) +
  ylim(c(0, 1)) + xlab("Distance") + ylab("Pr(Yea vote)") +
  ggtitle("Average case approach") + theme_bw()
p_sim
```



Observed value approach

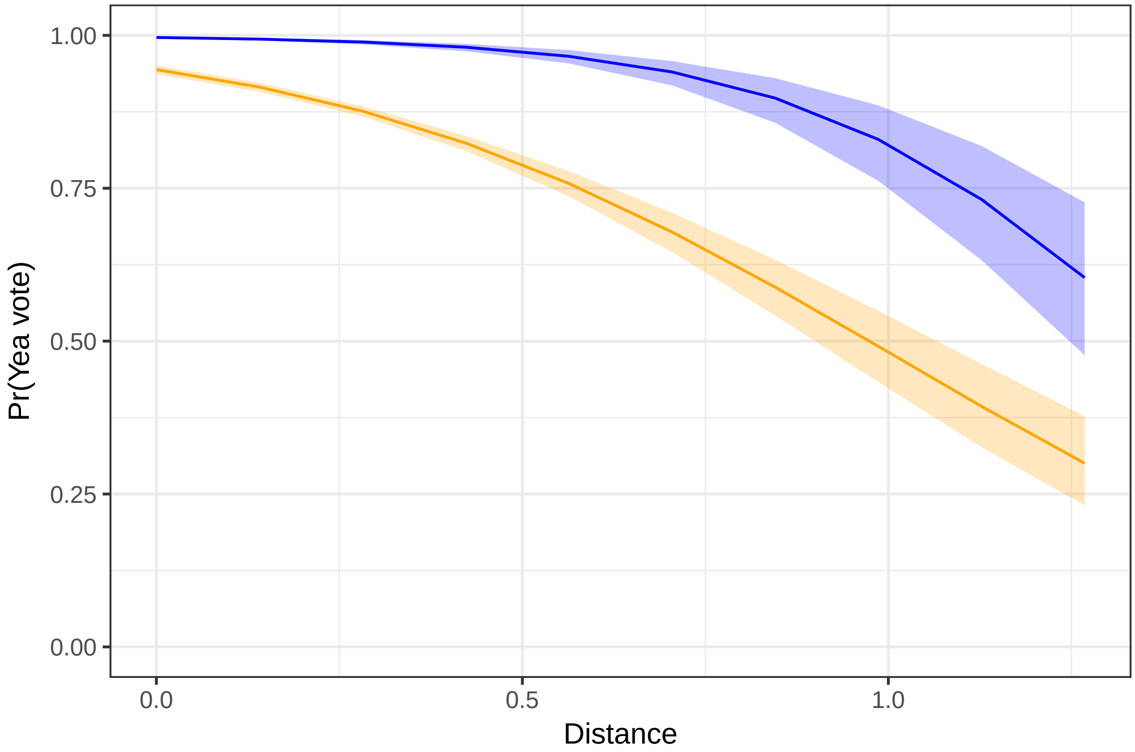
```
devtools::source_url("https://raw.githubusercontent.com/jkarreth/JKmisc/master/MCMC_observed_probs.R")

nom_xmat <- model.matrix(vote ~ lackqual + eucldist + strngprs + sameprty,
                          data = nom.dat)

nom_mcmc <- as.mcmc(nom.fit)
nom_mcmc_mat <- as.matrix(nom_mcmc)[, 1:ncol(nom_xmat)]
```

```r
nom_distance_sim <- seq(from = min(nom.dat$eucldist),
                        to = max(nom.dat$eucldist),
                        length.out = 10)

nom_obs_prob <- MCMC_observed_probs(model_matrix = nom_xmat,
                mcmc_out = nom_mcmc_mat,
                x_col = 3,
                x_range_vec = nom_distance_sim)

library(ggplot2)
p_obs <- ggplot(data = nom_obs_prob, aes(x = predictor, y = median_pp)) +
  geom_line() + geom_ribbon(aes(ymin = lower_pp, ymax = upper_pp), alpha = 0.25) +
  ylim(c(0, 1))  + xlab("Distance") + ylab("Pr(Yea vote)") +
  ggtitle("Observed value approach") + theme_bw()
p_obs
```



Comparing both approaches

```r
p_combined <- ggplot(data = nom_sim_prob, aes(x = predictor, y = median_pp)) +
  geom_line(color = "blue") + geom_ribbon(aes(ymin = lower_pp, ymax = upper_pp), fill = "blue", alpha =
  ylim(c(0, 1)) +
  geom_line(data = nom_obs_prob, aes(x = predictor, y = median_pp), color = "orange") +
  geom_ribbon(data = nom_obs_prob, aes(ymin = lower_pp, ymax = upper_pp), fill = "orange", alpha = 0.25
  xlab("Distance") + ylab("Pr(Yea vote)") + theme_bw()
p_combined
```


Pr(Yea vote)
1.00
0.75
0.50
0.25
0.00
0.0
0.5
1.0
Distance