Hierarchical Modeling of Gender and Community Effects on System Support

Jonathan Gilligan

Jun. 24, 2015

Important Note: This is preliminary work and I have not run full diagnostics on it, so do not take the results presented here as firm conclusions. They are preliminary observations, subject to change as I continue to work on this. However, the description of my methods and thinking may prove useful to the rest of the team.

Here I use an analysis of the "System Support" metric defined in Alfaro-Redondo, Vargas-Cullell & Seligson (2015), using questions 12.35–39 to illustrate some elementary aspects of multilevel modeling. I begin by presenting a simple example of using multilevel analysis without regression, which uses partial pooling to improve estimates of system support in communities for which we only have small sample sizes. Then I present an example ofof multilevel regression to estimate gender differences in system support, allowing for the gender difference to vary from commmunity to community.

1 Preparing data and simple descriptive statistics

First, we read in the SPSS MAR data and the questionnaire:

```
source('data_utils.R')
library(ggplot2)
library(RColorBrewer)


#
# Load MAR SPSS data and questionnaire
#
load_if_necessary()
```

Next, extract a data frame containing the questions for the system support index:

```
# Extract questions
q <- extract_q(questionnaire, Tab == 12 & Q %in% 35:39)
df <- extract_data_frame(mar_survey_data, q,
                         include_barcode = TRUE, short_barcode_name = TRUE,
                         include_gender = TRUE, include_site_no = TRUE)

# Change non-Likert responses to NA and Likert responses to integers
df <- filter_likert(df, q)
# Sort site numbers by fraction muslim at that site.
df <- fix_site_numbering(df, site_religion)
# Merge household and site religious characteristics into data frame
df <- merge(df, household_religion)
df <- merge(df, site_religion)

df <- df %>% mutate(hindu = hh_majority_religion == "Hindu")

site_labs <- site_labels_f_muslim(site_religion)
df$label <- ordered(df$site_no, levels = names(site_labs), labels = site_labs)

# Give short names for columns
names(df) <- sub('governance_accountability_', 'q', names(df))
names(df) <- sub('respondent_gender', 'gender', names(df))
```

Next, we generate the index: Turn 5 questions with answers in the range 1–5 into a single index that goes from 0–100:

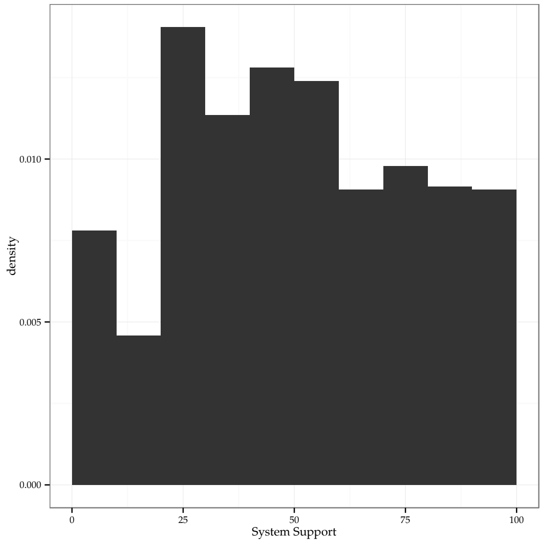


Figure 1: *Histogram of system support index.*

```
df <- df %>% mutate(s.sup = (q12.35 + q12.36 + q12.37 + q12.38 + q12.39 - 5) * 5)
N <- sum(!is.na(df$s.sup))
N.male <- sum(! is.na(df$s.sup) & df$gender == ’Male’)
N.female <- sum(! is.na(df$s.sup) & df$gender == ’Female’)
N.hindu <- sum(! is.na(df$s.sup)) & df$hindu
N.not.hindu <- sum(! is.na(df$s.sup)) & ! df$hindu
```

Now calculate descriptive statistics and make some plots so we can see how the new system support index is distributed:

```
summary(df$s.sup)
cat("Std. err of mean = ", sd(df$s.sup, na.rm=T) / sqrt(N), ’\n’, sep=’’)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA’s
    0.0    25.0    45.0    48.5    70.0   100.0     243
Std. err of mean = 0.877355
```

```
ggplot(df, aes(x=s.sup, y=..density..)) +
  geom_histogram(breaks = seq(0,100,10)) +
  labs(x = "System Support") +
  theme_bw(base_size = 10)
```

```
p <- ggplot(df, aes(x=s.sup, y=..density.., fill = gender)) +
  geom_histogram(breaks = seq(0,100,10), position=’dodge’) +
  facet_wrap(~label) +
  scale_fill_brewer(palette=’Dark2’) +
  labs(x = ’System Support’) +
  theme_bw(base_size = 10) +
  theme(axis.text.x = element_text(angle=90),
        strip.text = element_text(size=8))

p
```

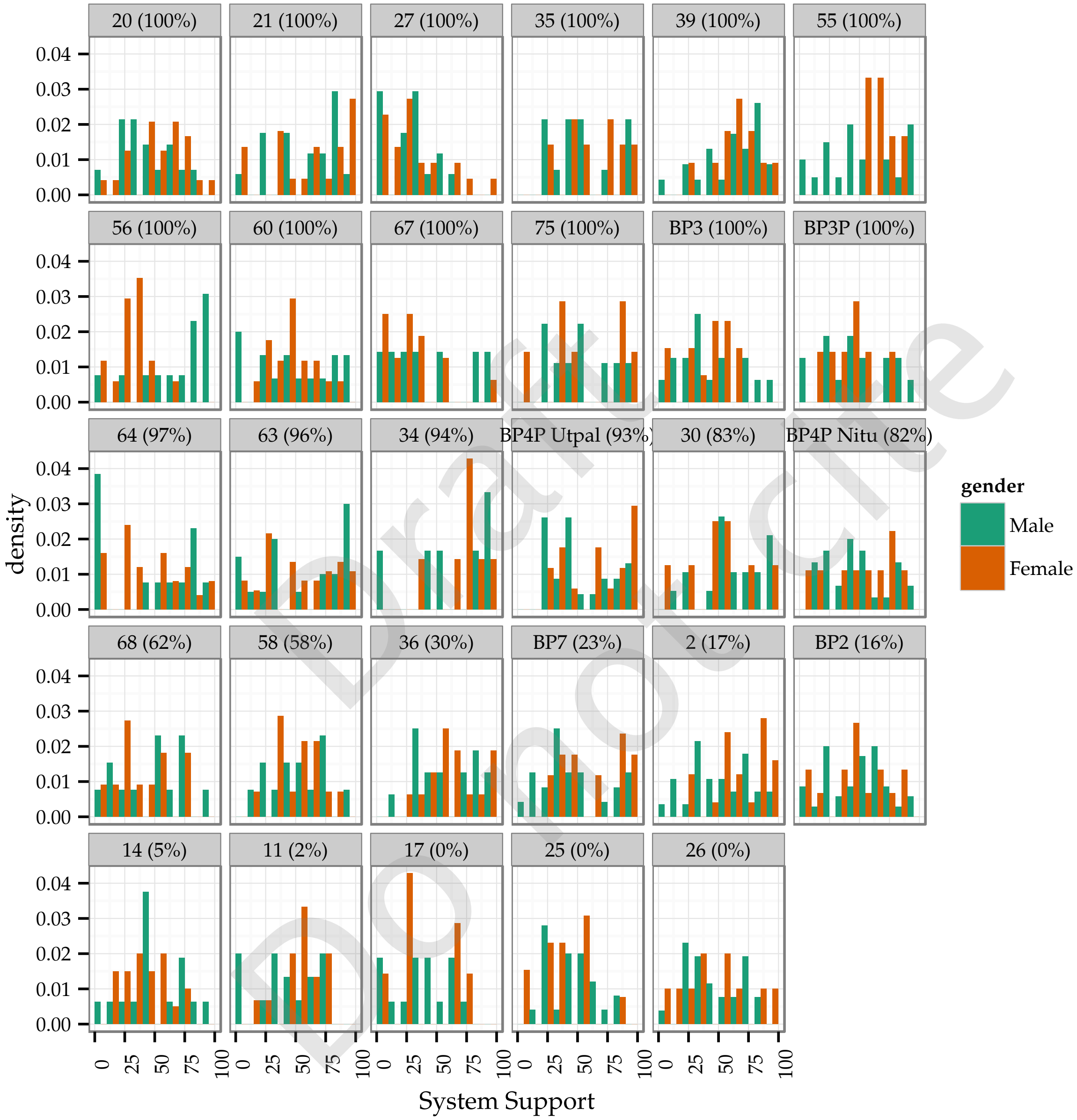


Figure 2: *Histogram of system support index, disaggregated by site number and gender. The captions above each sub-plot indicate the fraction of Muslim households among those surveyed at that site.*

2 Complete vs. partial pooling

In analyzing the data, we can use complete pooling, in which we lump all communities and all demographic groups together, giving us 961 measurements of the system support index. This is what we showed in Fig. 1. We could also analyze unpooled data, considering each site separately and males and females separately at each site.

The difficulty is that complete pooling has large numbers, so it gives a more precise estimate of the mean attitude of the whole population, but it doesn't tell us anything about differences between men and women or across communities. On the other hand, if we disaggregate by gender and site number, we know the system support index for an average of 16.6 people in each group and as few as 6, so our estimates of the statistical properties of each group will be very imprecise.

This analysis follows the model of Price, Nero & Gelman's (1996) analysis of radon concentrations in homes across Minnesota, as reported in and written up as a tutorial example in Gelman & Hill 2006.

Plot Figure 3a, showing the unpooled data for each gender at each site, with standard errors.

```
# complete pooling: state average of log(radon)
x <- df[! is.na(df$s.sup),]
N_SITES <- length(unique(x$site_no))
ybarbar <- mean(x$s.sup)
ylimits <- c(0,80)
y <- x %>% group_by(site_no, gender) %>%
  summarize(y = mean(s.sup), n = n(), se = sd(s.sup) / sqrt(n()))
y <- y %>% mutate(y.min = y - se, y.max = y + se, jittered = n + runif(1, -0.5, 0.5))
```

```
## Figure 3(a)
p3a <- ggplot(y, aes(x=jittered, ymin = y.min, ymax=y.max, y = y, color=gender)) +
  geom_pointrange() +
  scale_color_brewer(palette='Dark2') +
  geom_hline(yint=ybarbar) +
  ylim(ylimits) +
  labs(title="No Pooling", x = "Sample size at site", y = "System support") +
  theme_bw() +
  theme(legend.position=c(0.95,0.05), legend.justification=c(1,0))
print(p3a)
```

Initialize the seed for our random number generators:

```
seed.control = TRUE
if (seed.control) {
  seed <- 1405664174
} else {
  seed <- as.integer(Sys.time())
}
set.seed(seed)
num.stan.seeds <- 10
cur.stan.seed <- 1
stan.seeds <- sample.int(.Machine$integer.max, num.stan.seeds)
seed.message <- paste0("Seed = ", seed, " Stan Seeds = (", paste0(stan.seeds, collapse=', '), ")")
message(seed.message)
```

```
[1] "Seed = 1405664174 Stan Seeds = (167155411, 1917337316, 1479868761,"
[2] "837399473, 739607561, 538596304, 1521173651, 1282463969, 214793702,"
[3] "1735528271)"
```

Now, calculate the partially pooled means and standard errors. Here is a the code for our model, which we keep in the file `system_supt_multilevel_gender_nopred.stan`:

```
data {
  int<lower=0> N;
  int<lower=0> N_SITES;
  int<lower=0,upper=N_SITES> site[N];
  int<lower=0,upper=1> i_female[N];
  real<lower=0,upper=100> s[N];
}
```

```
parameters {
  real<lower=0,upper=100> mu_0;
  real<lower=0> sigma_0;
  real<lower=0> sigma_c;
  vector[N_SITES] mu_c[2];
}

model {
  mu_0 ~ normal(50., 25.);
  sigma_0 ~ gamma(25., 1.0);
  sigma_c ~ gamma(20.0, 1.0);
  for(i in 1:2) {
    mu_c[i] ~ normal(mu_0, sigma_0);
  }
  for(i in 1:N) {
    s[i] ~ normal(mu_c[i_female[i]+1][site[i]], sigma_c);
  }
}
```

To use this model, first we prepare the data:

```
library(rstan)
library(coda)

rstan_options(auto_write = TRUE)
options(mc.cores = min(4, parallel::detectCores()))

# prepare a list of data to pass to Stan model
s.sup.gender.data <- list (N = N, N_SITES = N_SITES, s = x$s.sup,
                           site = unclass(x$site_no),
                           i_female = ifelse(x$gender == "Female", 1, 0))
# Tell Stan the names of the parameters it should report back to us
s.sup.parameters <- c ("mu_c", "mu_0", "sigma_0", "sigma_c")
```

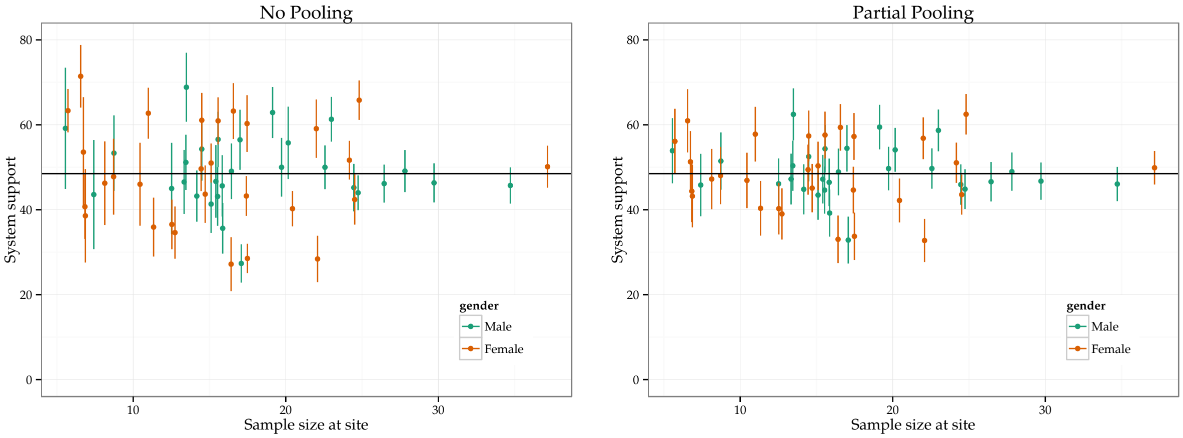
Now we compile a model and run it:

```
# Compile the Stan model, initializing the data and parameters
s.sup.gender.nopred.model <- stan_model("system_supt_multilevel_gender_nopred.stan",
                                "Partial Pooling, No Prediction")
s.sup.gender.stanfit <- sampling(s.sup.gender.nopred.model, data = s.sup.gender.data,
                          pars = s.sup.parameters,
                          chains = 4,iter = 10000,
                          seed = stan.seeds[cur.stan.seed])
saveRDS(s.sup.gender.stanfit, "s_sup_gender_stanfit.Rds")
cur.stan.seed <- cur.stan.seed + 1
if (cur.stan.seed > num.stan.seeds) cur.stan.seed <- 1
```

Next, we process the output of the model run: We take the samples from the Markov-Chain Monte Carlo sampler and calculate means and standard errors:

```
tmp <- as.array(s.sup.gender.stanfit)
post.nopred <- do.call(rbind, lapply(1:4, function(i) tmp[,i,]))

mean.a.nopred <- matrix(NA, nrow=N_SITES, ncol = 2)
sd.a.nopred <- matrix(NA, nrow = N_SITES, ncol = 2)
for (i in 1:N_SITES) {
  for (j in 1:2) {
  mean.a.nopred[i,j] <- mean(post.nopred[ , paste(’mu_c[’,j,’,’,i,’]’, sep=’’)])
  sd.a.nopred[i,j] <-     sd(post.nopred[ , paste(’mu_c[’,j,’,’,i,’]’, sep=’’)])
  }
}
colnames(mean.a.nopred) <- c(’Male’,’Female’)
colnames(sd.a.nopred) <- c(’Male’,’Female’)
mean.a.nopred <- as.data.frame(mean.a.nopred)
sd.a.nopred <- as.data.frame(sd.a.nopred)
mean.a.nopred$site_no <- levels(x$site_no)[1:N_SITES]
sd.a.nopred$site_no <- levels(x$site_no)[1:N_SITES]
mean.a.nopred <- mean.a.nopred %>% gather(key=’gender’, value=’y’, -site_no)
sd.a.nopred <- sd.a.nopred %>% gather(key="gender", value="se", -site_no)
mean.a.nopred$n <- NA
```



(a) *No-pooling site-level analysis.*

(b) *Multilevel (partial pooling) analysis.*

Figure 3: *Estimates ± standard errors for the average system support at MAR sites plotted versus the (jittered) number of people of each gender interviewed at each site. In both analyses, no effects of specific characteristics at the site-level or individual-level effects are considered. The sites with fewer measurements have more variable estimates with higher standard errors. The horizontal line in each figure shows an estimate of average across both genders and all MAR sites (complete pooling of all sites). Fig. 3a illustrates a problem with the no-pooling analysis: it systematically causes us to think that certain sites are more extreme, just because they have smaller sample sizes.*

```
mean.a.nopred$jittered <- NA
for (i in 1:nrow(mean.a.nopred)) {
  mean.a.nopred$n[i] <-
    y$n[y$site_no == mean.a.nopred$site_no[i] &
          y$gender == mean.a.nopred$gender[i]]
  mean.a.nopred$jittered[i] <-
    y$jittered[y$site_no == mean.a.nopred$site_no[i] &
                y$gender == mean.a.nopred$gender[i]]
}
```

Finally, we plot the data:

```
  ## Figure 3(b)
  frame3b <- merge(mean.a.nopred, sd.a.nopred)
  frame3b <- frame3b %>% mutate( y.min = y - se, y.max = y + se)
  p3b <- ggplot(frame3b, aes(x=jittered, ymin = y.min,
                             ymax=y.max, y = y, color=gender)) +
    geom_pointrange() +
    scale_color_brewer(palette='Dark2') +
    geom_hline(yint=ybarbar) +
    ylim(ylimits) +
    labs(title="Partial Pooling", x = "Sample size at site",
         y = "System support") +
    theme_bw() +
    theme(legend.position=c(0.95,0.05), legend.justification=c(1,0))
  print(p3b)
```

3 Hierarchical Regressions: Mixed-Effects Models

In classical regression, we treat effects as either fixed or random. Gelman & Hill (2006, pp. 244–246) point out that statisticians use the term “fixed effects” to mean at least five different and inconsistent things, and argue that this indicates a fundamental ambiguity about the concept even among experts, but what all the definitions seem to be to be getting at in one way or another is that fixed effects differ from random effects in terms of the completeness of information the experimenter believes she has: Different definitions describe fixed effects model as applying when one thinks she has sampled the entire population of interest, when she thinks she has measured all possible values of a factor, or when she thinks that the factor of interest has the same effect on all individuals. Conversely, random effects imply greater ignorance and uncertainty: they apply when one has sampled only a small fraction of the population of interest, when one has sampled only a small number of the possible values of an independent variable, or when one believes that the effect of a factor is randomly different from one individual to another.

One way of looking at gender in our data is that we have measured all possible values[1] in 442 female subjects and 519 male ones. If we believed that gender has the same effect on everyone, this would be a very large sample, and even with random noise in the samples we should be able to measure the effect of gender very well unless it is very small.

However, there is another possibility: We have sampled men and women who belong to different religions and live in different communities and who have different socioeconomic status within those communities. If the effect of gender depends on these other variables, then we do not have a large sample for each possible variation of these factors. Thus, gender takes on some aspects of a random variable, for which we have drawn a small sample from a large popuation and only measured a small fraction of the possible combinations of all the different factors at play.

However, we have strong theoretical reasons not to belive that gender effects are completely random. Despite variation at the community level, and across religions and socioeconomic strata, we expect some commonality. Thus, we draw on mixed-effects modeling and specifically a hierarchical (also called multilevel) formulation.

Let us suppose that if there were no gender differences, there would still variation in system support within each community. For simplicity, let us assume that this variation follows a normal distribution defined by a mean and standard deviation $\mu_{c,i}$ and $\sigma_{c,i}$ that characterize the community at site i. The parameter $\mu_{c,i}$ represents the average attitude in that community and $\sigma_{c,i}$ represents the range of attitudes (small σ means a narrow range and a large σ would mean a large range). We surveyed a random sample from each site, so we expect our answers to be drawn at random from the normal distribution of opinions within that community. Mathematically, if s_j represents the system support attitude of person j at site i, we write this as

$$s_j \sim \text{normal}(\mu_{c,i}, \sigma_{c,i}), \tag{1}$$

where $x \sim y$ represents drawing a value for x at random from probability distribution y.

Each community is different, but they still have a lot in common, so we might hypothesize that the set of communities that our survey studied were drawn from a larger pool of communities whose distribution of μ_c form a normal distribution characterized by what we call hyper-parameters: hyper-mean μ_0 and hyper-standard deviation σ_0. The hyper-parameter μ_0 would represent the average attitude across all of the communities and σ_0 would represent the variation in attitude from one community to another. There could be further hyperparemters characterizing the distribution of σ_c across communities (e.g., variation in the amount of agreement or disagreement within the community about support for the system), but in the interest of brevity and simplicity, for this analysis I will assume that σ_c is the same for all communities.[2]

One way to interpret this is to say that μ_0 represents what the communities have in common and σ_0 represents the differences among them.

If we had chosen our communities at random, we would expect the community-level means μ_c to be distributed like random draws from "parent" distributions. In fact, we did not choose our communities

[1] In fact, we have not: we do not consider Hijras or others who do not conform to the traditional gender binary, but in the context of this study in rural southwestern Bangladesh, it seems reasonable to treat gender as though it only has two possible values.

[2] If you want to investigate the possibility of different degrees of within-community variation, the hyperdistribution of σ_c often follows a Gamma distribution. There is an excellent, clear, and practically oriented treatment of such hyperdistributions in John Kruschke's great textbook, *Doing Bayesian Data Analysis* (Kruschke 2014). I enthusiastically recommend this book to anyone interested in the subject at a beginner level. For more advanced treatments, see Gelman *et al.* (2013) and Gelman & Hill (2006)

at random, and this complicates our analysis but I will sweep this difficulty under the carpet for the purposes of this treat the community-level parameters as though they were in fact drawn at random. The consequence of this unjustified assumption is that we cannot use this statistical analysis to draw inferences from these communities to other communities that do not fit the parent distribution we infer from these communities.

So we now have a model for public opinion regarding support for the system of government: The whole group of MAR and BP sites is characterized by hyperparameters, such as μ_0 and σ_0. Each site has its own value of $\mu_{c,i}$, which we assume are drawn at random from the normal parent distribution characterized by μ_0 and σ_0. The individuals in the community are drawn at random from the population of the community, whose attitudes we assume follow a normal distribution characterized by $\mu_{c,i}$ and σ_c (I write σ_c instead of $\sigma_{c,i}$ to indicate that I have chosen a model in which σ_c does not vary across communities).

Mathematically, we can write this as

$$\mu_{c,i} \sim \text{normal}(\mu_0, \sigma_0) \tag{2}$$

$$s_j \sim \text{normal}(\mu_{c,i}, \sigma_c) \tag{3}$$

That is essentially the analysis I performed in Section 2. In that section, I treated gender as a completely random effect, which meant that, in effect, men and women formed separate communities at each site so, for instance, I treated Site #30 as though it were two separate and unrelated sites, one of which comprised all the men at Site #30, and the other of which comprised the women.

When we speak of “hierarchical modeling” or “multilevel modeling,” this refers to the hierarchy of probability distributions: We treat individuals as though they are drawn at random from a community-level probability distribution, and we treat the communities as though they are drawn at random from a parent distribution (sometimes called a hyperdistribution).

3.1 Modeling gender effects

Now we can turn to gender effects. On top of the model presented in the previous section, it is reasonable to believe that within each community there is a variety of gender differences: not every woman has the same opinion and neither does every man. For simplicity let us assume that there is a normal distribution of differences between men and women in the community characterized by a mean gender difference $\mu_{g,i}$ at Site i, so is s_j represents the system support attitude of individual j at site i,

$$s_j \sim \text{normal}(\mu_{c,i} + \mu_{g,i} \cdot I_{\text{gender},j}, \sigma_c), \tag{4}$$

where $I_{\text{gender},j}$ is a dichotomous indicator variable for the gender of person j. To put this equation in the context of the multi-level model, we can write

$$\mu_{c,i} \sim \text{normal}(\mu_0, \sigma_0) \tag{5}$$

$$\mu_{g,i} \sim \text{normal}(\mu_{g0}, \sigma_{g0}) \tag{6}$$

$$s_j \sim \text{normal}(\mu_{c,i} + \mu_{g,i} \cdot I_{\text{gender},j}, \sigma_c), \tag{7}$$

where μ_{g0} and σ_{g0} are hyperparameters that characterize the distribution of gender differences across communities.

We have to make some further assumptions about what μ_0, σ_0, σ_c, μ_{g0}, and σ_{g0} might be, but since we don’t have much previous empirical evidence to go on, we make very loose assumptions and leave room for them to take on a wide range of possible values, which we will then constrain by fitting them to the data.

Note how many places in this analysis I have to make assumptions about what distributions or hyperdistributions to use. In the end, a good analysis will go back and check its results against the data: Do the distributions predicted by the final model agree with the observed distribution of data or not? There is a danger of circularity in fitting the model to the data and then asking how well it fits, but if you do this well and honestly, if you try to fit a “square” model, so to speak, to a “circular” distribution of data, you will see “corners” poking out. More concretely, if data follows a parabola and you try to fit it with a straight line, you will see that the data shows curvature that your model does not. This is one reason why statistics books emphasize “degrees of freedom” so much: If the data has enough degrees of freedom remaining after fitting a model, then any fundamental mismatch between the data and the model will be apparent.

Now we are ready to write code and fit our model.

Here is my model of gender effects:

```
data {
  int<lower=0> N;
  int<lower=1> N_SITES;
  int<lower=1,upper=N_SITES> site[N];
  int<lower=0,upper=1> i_female[N];
  real<lower=0,upper=100> s[N];
}

transformed data {
  int<lower=1,upper=2> i_gender[N];

  for(i in 1:N) {
    i_gender[i] <- i_female[i] + 1;
  }
}

parameters {
  real<lower=0,upper=100> mu_0;
  real<lower=0> sigma_0;
  real mu_gender_0;
  real<lower=0> sigma_gender_0;
  vector[N_SITES] mu_c;
  vector[N_SITES] mu_gender_c;
  vector<lower=0>[N_SITES] sigma_c;
}

transformed parameters {
  vector [N_SITES] a[2];

  // symmetric treatment of gender effect to avoid collinearity
  // positive gender effect = females support government more than men.
  a[1] <- mu_c - 0.5 * mu_gender_c;
  a[2] <- mu_c + 0.5 * mu_gender_c;
}

model {
  mu_0 ~ normal(50., 50.);
  mu_gender_0 ~ normal(0.,30.);
  sigma_0 ~ gamma(10., 1.0);
  sigma_gender_0 ~ gamma(10.,1.0);
  mu_c ~ normal(mu_0, sigma_0);
  sigma_c ~ gamma(30.0, 1.0);
  mu_gender_c ~ normal(mu_gender_0, sigma_gender_0);
  for(i in 1:N) {
    s[i] ~ normal(a[i_gender[i]][site[i]], sigma_c[site[i]]);
  }
}
```

First, we compile and run the model:

```
s.sup.gender.model <- stan_model("system_supt_multilevel_gender_1.stan",
                                 "Partial Pooling, Gender Prediction")
s.sup.gender.stanfit <- sampling(s.sup.gender.model, data = s.sup.gender.data,
                                 chains = 4, iter = 10000,
                          seed = stan.seeds[cur.stan.seed])
cur.stan.seed <- cur.stan.seed + 1
if (cur.stan.seed > num.stan.seeds) cur.stan.seed <- 1
```

Next, just as in the previous example, we convert the output to a useful format:

```
post.gender <- as.matrix(s.sup.gender.stanfit)
```

Now we have to translate the parameter names from computer variable names to something that makes sense to people:

```
param_x <- data.frame(Parameter = names(s.sup.gender.stanfit), Label = NA,
                      stringsAsFactors=FALSE)
param_x <- param_x %>% filter(grepl('mu_gender_c', Parameter, fixed=T))
for(i in 1:nrow(param_x)) {
  param_x$Label[i] <- levels(y$site_no)[i]
  param_x$frac_muslim[i] <- site_religion$frac_muslim[site_religion$site_no == param_x$Label[i]]
}
param_x$Label <- as.character(param_x$Label)
param_names <- param_x %>% select(Parameter, Label)

pn_x <- data.frame(Parameter = 'mu_gender_0', Label = 'Overall average', stringsAsFactors=FALSE)
param_names <- rbind(param_names, pn_x)
param_x$mu <- unlist(lapply(param_x$Parameter, function(x) {
  mean(post.gender[,as.character(x)])}
  ))
site_56_mean <- param_x$mu[param_x$Label == '56']

param_x <- param_x$Label[order(param_x$frac_muslim)]
param_x <- data.frame(Parameter = param_x, value = 1:length(param_x), stringsAsFactors = FALSE)
param_x <- rbind(param_x, data.frame(Parameter = 'Overall average', value = 0))
```

To get a clearer sense of how our estimates of gender effects vary across communities, we plot the highest-density intervals of the posterior probability distributions for the gender-effect parameter at each community ($\mu_{g,i}$) and overall (μ_{g0}):

```
loose_ci<- 0.05
tight_ci  <- 0.001
loose_pct <- 100 * (1 - loose_ci)
tight_pct <- 100 * (1 - tight_ci)
```

```
library(ggmcmc)
```

```
Loading required package: GGally

Attaching package: 'GGally'

The following object is masked from 'package:dplyr':

    nasa
```

```
gs <- ggs(As.mcmc.list(s.sup.gender.stanfit), 'mu_gender', par_labels = param_names)
cat_plot1 <- ggs_caterpillar(gs, line = 0, X = param_x,
          thick_ci = c(loose_ci/2, 1 - loose_ci/2),
          thin_ci = c(tight_ci/2, 1 - tight_ci/2)) +
  scale_x_continuous(breaks = seq(-30,25,10)) +
  scale_y_continuous(breaks = param_x$value, labels = param_x$Parameter) +
  labs(x = "Gender Effect", y = "Site") + theme_bw()
plot(cat_plot1)
```

The upshot is that there is not a significant gender effect across communities and even within communities, only one of our 29 sites shows significant effects with 95% certainty (I am using the language of 95% certainty rather than $p < 0.05$ because I am working with Bayesian posterior probability distributions rather than null hypothesis significance tests). Since we are performing 30 significance tests, there is a 79% probability of getting at least one Type-I error at the 95% confidence level, but it is striking that our analysis gives us a much higher degree of confidence: Based on the posterior probability distribution from our analysis, we find 99.9% probability that the gender differences at site 56 are nonzero, and if we take account of the many comparisons we are making (29 sites plus the overall effect for all sites) there is only a 3% chance that we would see a result this large just by happenstance. However, it is important to consider the "garden of forking paths" (Gelman & Loken 2014), where contingent decisions about how to analyze data can lead analysts to severely underestimate the number of multiple comparisons that go into an analysis and consequently to severely underestimate the probability that a result arose purely by chance.

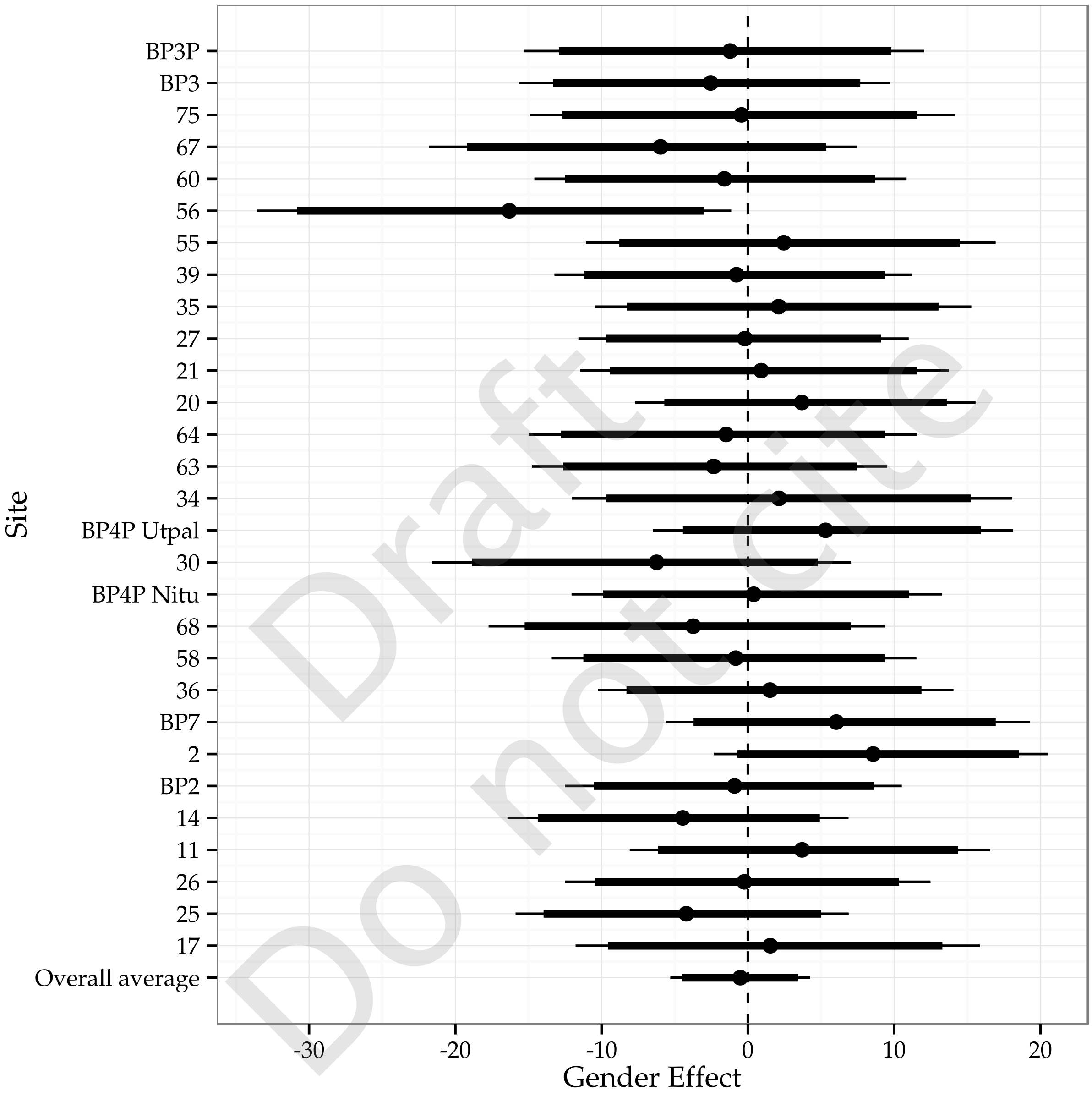


Figure 4: *Gender effects: The vertical dashed line corresponds to no effect. Positive (right of the dashed line) means women support the system more than men do. The thick bars represent 95% confidence intervals (to be precise, they represent the highest-density intervals of the posterior probabilty distributions of the parameters) and the thin lines represent 99.9% confidence intervals.*

It is equally, if not more important, to look at the size of an effect than simply the probablity that it is not zero. Finding high statistical significance in an effect too small to have any practical impact would not be a useful finding. From this perspective, site 56 is very interesting because as Fig. 4 shows, although the size of the gender difference at Site 56 is very uncertain, its most likely value is 17 points on a 100 point scale, where the average attitude toward government across all sites was 48.4, so there is considerable probability that differences between genders at this site may be roughly one third of the total score.

Thus, while this analysis does not allow us to estimate the size of this effect precisely, and while we that shows up strongly only at a single site, this analysis does give us evidence that the effect of gender *could* be very large and with guidance from theory about the mechanisms of gender effects and their interactions with other factors, such as religious minority status and poverty, we might be able to produce clearer and more certain estimates of gender effects at Site 56 and other sites. This line of thinking follows the recommendations of Kruschke (2014), Kruschke (2013), and Gelman & Stern (2006) in focusing on posterior distributions of effect size (e.g., Kruschke's "Region of Practical Equivalence") rather than on rejecting a point null hypothesis.

4 Analysis by Religion

Now we repeat the whole analysis, but using an indicator for Hindu households as the variable of interest:

First plot Figure 5a, showing the unpooled data for each religion at each site, with standard errors.

```
# complete pooling: state average of log(radon)
x <- df[! is.na(df$s.sup),]
N_SITES <- length(unique(x$site_no))
ybarbar <- mean(x$s.sup)
ylimits <- c(0,80)
y <- x %>% group_by(site_no, hindu) %>%
  summarize(y = mean(s.sup), n = n(), se = sd(s.sup) / sqrt(n()))
y <- y %>% mutate(y.min = y - se, y.max = y + se, jittered = n + runif(1, -0.5, 0.5))
```

```
## Figure 5(a)
p5a <- ggplot(y, aes(x=jittered, ymin = y.min, ymax=y.max, y = y, color=hindu)) +
  geom_pointrange() +
  scale_color_brewer(palette='Dark2', name = "Hindu?", labels=c("No", "Yes")) +
  geom_hline(yint=ybarbar) +
  ylim(ylimits) +
  labs(title="No Pooling", x = "Sample size at site", y = "System support") +
  theme_bw() +
  theme(legend.position=c(0.95,0.05), legend.justification=c(1,0))
print(p5a)
```

Now, calculate the partially pooled means and standard errors. Here is a the code for our model, which we keep in the file `system_supt_multilevel_hindu_nopred.stan`:

To use this model, first we prepare the data:

```
# prepare a list of data to pass to Stan model
s.sup.hindu.data <- list (N = N, N_SITES = N_SITES, s = x$s.sup,
                          site = unclass(x$site_no),
                          i_hindu = x$hindu)
# Tell Stan the names of the parameters it should report back to us
s.sup.parameters <- c ("mu_c", "mu_0", "sigma_0", "sigma_c")
```

Now we compile a model and run it:

```
# Compile the Stan model, initializing the data and parameters
s.sup.hindu.nopred.model <- stan_model("system_supt_multilevel_hindu_nopred.stan",
                                "Partial Pooling, No Prediction")
s.sup.hindu.stanfit <- sampling(s.sup.hindu.nopred.model, data = s.sup.hindu.data,
                         pars = s.sup.parameters,
                         chains = 4,iter = 10000,
                         seed = stan.seeds[cur.stan.seed])
```

```
saveRDS(s.sup.hindu.stanfit, "s_sup_hindu_stanfit.Rds")
cur.stan.seed <- cur.stan.seed + 1
if (cur.stan.seed > num.stan.seeds) cur.stan.seed <- 1
```

Next, we process the output of the model run: We take the samples from the Markov-Chain Monte Carlo sampler and calculate means and standard errors:

```
tmp <- as.array(s.sup.hindu.stanfit)
post.nopred <- do.call(rbind, lapply(1:4, function(i) tmp[,i,]))

mean.a.nopred <- matrix(NA, nrow=N_SITES, ncol = 2)
sd.a.nopred <- matrix(NA, nrow = N_SITES, ncol = 2)
for (i in 1:N_SITES) {
  for (j in 1:2) {
  mean.a.nopred[i,j] <- mean(post.nopred[ , paste('mu_c[',j,',',i,']', sep='')])
  sd.a.nopred[i,j] <-     sd(post.nopred[ , paste('mu_c[',j,',',i,']', sep='')])
  }
}
colnames(mean.a.nopred) <- c('Other','Hindu')
colnames(sd.a.nopred) <- c('Other','Hindu')
mean.a.nopred <- as.data.frame(mean.a.nopred)
sd.a.nopred <- as.data.frame(sd.a.nopred)
mean.a.nopred$site_no <- levels(x$site_no)[1:N_SITES]
sd.a.nopred$site_no <- levels(x$site_no)[1:N_SITES]
mean.a.nopred <- mean.a.nopred %>% gather(key='hindu', value='y', -site_no)
sd.a.nopred <- sd.a.nopred %>% gather(key="hindu", value="se", -site_no)
mean.a.nopred$hindu <- mean.a.nopred$hindu == 'Hindu'
sd.a.nopred$hindu <- sd.a.nopred$hindu == 'Hindu'
mean.a.nopred$n <- NA
mean.a.nopred$jittered <- NA
for (i in 1:nrow(mean.a.nopred)) {
  index <- which(y$site_no == mean.a.nopred$site_no[i] &
          y$hindu == mean.a.nopred$hindu[i])
  if (length(index) > 0) {
    mean.a.nopred$n[i] <-y$n[index]
    mean.a.nopred$jittered[i] <- y$jittered[index]
  }
}
```

Finally, we plot the data:

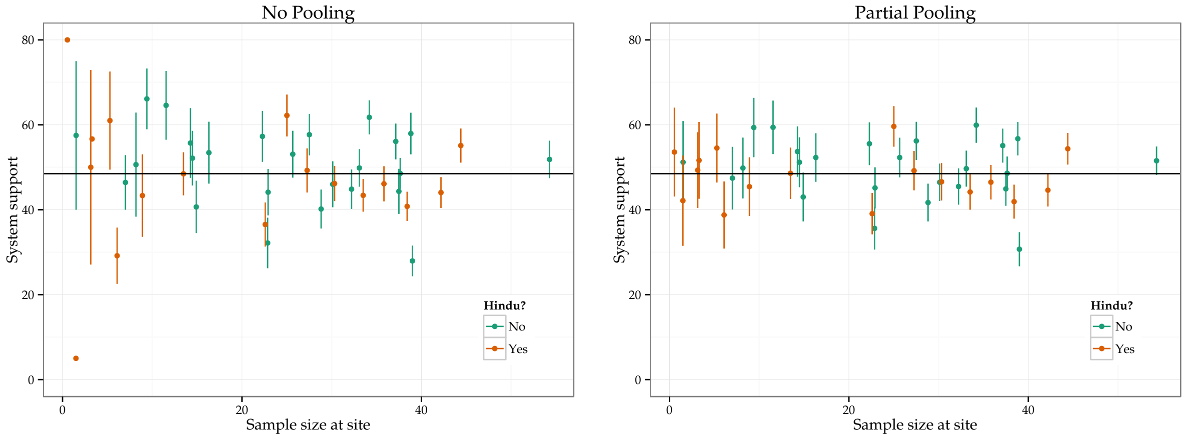
```
  ## Figure 5(b)
  frame5b <- merge(mean.a.nopred, sd.a.nopred)
  frame5b <- frame5b %>% mutate( y.min = y - se, y.max = y + se)
  p5b <- ggplot(frame5b, aes(x=jittered, ymin = y.min,
                            ymax=y.max, y = y, color=hindu)) +
    geom_pointrange() +
    scale_color_brewer(palette='Dark2', name = "Hindu?", labels = c("No", "Yes")) +
    geom_hline(yint=ybarbar) +
    ylim(ylimits) +
    labs(title="Partial Pooling", x = "Sample size at site",
         y = "System support") +
    theme_bw() +
    theme(legend.position=c(0.95,0.05), legend.justification=c(1,0))
  print(p5b)
```

5 Hierarchical Regressions: Mixed-Effects Models

As before, but regressing against whether the household is Hindu.

Here is my model of the effect of being Hindu:

```
data {
  int<lower=0> N;
  int<lower=1> N_SITES;
  int<lower=1,upper=N_SITES> site[N];
  int<lower=0,upper=1> i_hindu[N];
  real<lower=0,upper=100> s[N];
```



(a) *No-pooling site-level analysis.*

(b) *Multilevel (partial pooling) analysis.*

Figure 5: *Estimates ± standard errors for the average system support at MAR sites plotted versus the (jittered) number of people of each religion interviewed at each site. In both analyses, no effects of specific characteristics at the site-level or individual-level effects are considered. The sites with fewer measurements have more variable estimates with higher standard errors. The horizontal line in each figure shows an estimate of average across both religions and all MAR sites (complete pooling of all sites). Fig. 3a illustrates a problem with the no-pooling analysis: it systematically causes us to think that certain sites are more extreme, just because they have smaller sample sizes.*

```
}

transformed data {
  int<lower=1,upper=2> i_religion[N];

  for(i in 1:N) {
    i_religion[i] <- i_hindu[i] + 1;
  }
}

parameters {
  real<lower=0,upper=100> mu_0;
  real<lower=0> sigma_0;
  real mu_religion_0;
  real<lower=0> sigma_religion_0;
  vector[N_SITES] mu_c;
  vector[N_SITES] mu_religion_c;
  vector<lower=0>[N_SITES] sigma_c;
}

transformed parameters {
  vector [N_SITES] a[2];

  // symmetric treatment of religion effect to avoid collinearity
  // positive religion effect = hindus support government more than men.
  a[1] <- mu_c - 0.5 * mu_religion_c;
  a[2] <- mu_c + 0.5 * mu_religion_c;
}

model {
  mu_0 ~ normal(50., 50.);
  mu_religion_0 ~ normal(0.,30.);
  sigma_0 ~ gamma(10., 1.0);
  sigma_religion_0 ~ gamma(10.,1.0);
  mu_c ~ normal(mu_0, sigma_0);
  sigma_c ~ gamma(30.0, 1.0);
  mu_religion_c ~ normal(mu_religion_0, sigma_religion_0);
  for(i in 1:N) {
    s[i] ~ normal(a[i_religion[i]][site[i]], sigma_c[site[i]]);
  }
}
```

First, we compile and run the model:

```
s.sup.hindu.model <- stan_model("system_supt_multilevel_hindu_1.stan",
                                "Partial Pooling, Religion Prediction")
s.sup.hindu.stanfit <- sampling(s.sup.hindu.model, data = s.sup.hindu.data,
                                chains = 4, iter = 10000,
                         seed = stan.seeds[cur.stan.seed])
cur.stan.seed <- cur.stan.seed + 1
if (cur.stan.seed > num.stan.seeds) cur.stan.seed <- 1
```

Next, just as in the previous example, we convert the output to a useful format:

```
post.hindu <- as.matrix(s.sup.hindu.stanfit)
```

Now we have to translate the parameter names from computer variable names to something that makes sense to people:

```
param_x <- data.frame(Parameter = names(s.sup.hindu.stanfit), Label = NA,
                         stringsAsFactors=FALSE)
param_x <- param_x %>% filter(grepl('mu_religion_c', Parameter, fixed=T))
for(i in 1:nrow(param_x)) {
  param_x$Label[i] <- levels(y$site_no)[i]
  param_x$frac_muslim[i] <- site_religion$frac_muslim[site_religion$site_no == param_x$Label[i]]
}
param_x$Label <- as.character(param_x$Label)
param_names <- param_x %>% select(Parameter, Label)

pn_x <- data.frame(Parameter = 'mu_religion_0', Label = 'Overall average', stringsAsFactors=FALSE)
param_names <- rbind(param_names, pn_x)
param_x$mu <- unlist(lapply(param_x$Parameter, function(x) {
  mean(post.hindu[,as.character(x)])}
  ))
site_56_mean <- param_x$mu[param_x$Label == '56']

param_x <- param_x$Label[order(param_x$frac_muslim)]
param_x <- data.frame(Parameter = param_x, value = 1:length(param_x), stringsAsFactors = FALSE)
param_x <- rbind(param_x, data.frame(Parameter = 'Overall average', value = 0))
```

To get a clearer sense of how our estimates of religion effects vary across communities, we plot the highest-density intervals of the posterior probability distributions for the religion-effect parameter at each community ($\mu_{g,i}$) and overall (μ_{g0}):

```
gs <- ggs(As.mcmc.list(s.sup.hindu.stanfit), 'mu_religion', par_labels = param_names)
cat_plot2 <- ggs_caterpillar(gs, line = 0, X = param_x,
          thick_ci = c(loose_ci/2, 1 - loose_ci/2),
          thin_ci = c(tight_ci/2, 1 - tight_ci/2)) +
  scale_x_continuous(breaks = seq(-30,25,10)) +
  scale_y_continuous(breaks = param_x$value, labels = param_x$Parameter) +
  labs(x = "Religion Effect", y = "Site") + theme_bw()
plot(cat_plot2)
```

References

ALFARO-REDONDO, RONALD, JORGE VARGAS-CULLELL & MITCHELL A. SELIGSON (2015). *Political Culture in Costa Rica: Long-term slide continues in attitudes favoring stable democracy*. Tech. rep. Latin American Public Opinion Project.

GELMAN, ANDREW, & JENNIFER HILL (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press. 654 pp.

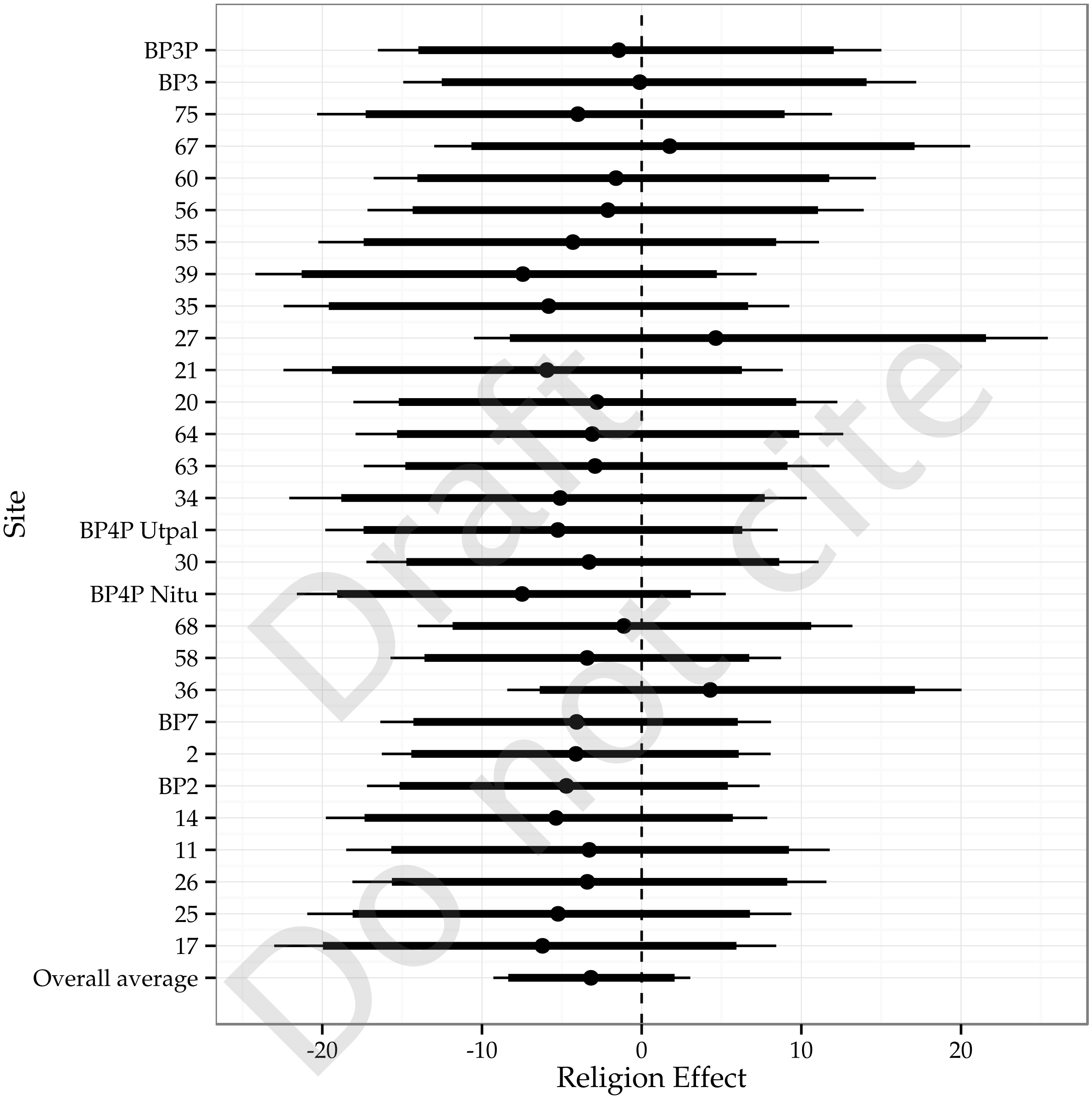


Figure 6: *Religion effects: The vertical dashed line corresponds to no effect. Positive (right of the dashed line) means women support the system more than men do. The thick bars represent 95% confidence intervals (to be precise, they represent the highest-density intervals of the posterior probabilty distributions of the parameters) and the thin lines represent 99.9% confidence intervals.*

Gelman, Andrew, & Eric Loken (2014). “The Statistical Crisis in Science”. *American Scientist* 102.6, p. 460. doi: 10.1511/2014.111.460.

Gelman, Andrew, & Hal Stern (2006). “The difference between “significant” and “not significant” is not itself statistically significant”. *The American Statistician* 60.4, pp. 328–331.

Gelman, Andrew, *et al.* (2013). *Bayesian Data Analysis, Third Edition*. CRC Press. 677 pp.

Kruschke, John K. (2013). “Bayesian estimation supersedes the t test”. *Journal of Experimental Psychology: General* 142, pp. 573–603. doi: 10.1037/a0029146.

Kruschke, John K. (2014). *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*. Avademic Press.

Price, Phillip N., Anthony V. Nero & Andrew Gelman (1996). “Bayesian prediction of mean indoor radon concentrations for Minnesota counties”. *Health Physics* 71, pp. 922–936.