MRP example: self-reported abortion in Uganda

Monica Alexander

21/04/2021

Contents

1 Overview

This document goes through a worked example of multilevel regression and post-stratification to get estimates of self-reported abortion incidence in Uganda. The data used are the 2018 PMA survey and the 2014 Census. Both datasets were obtained through IPUMS.

2 Load in data and tidy up

Load in the packages:

```
library(tidyverse)
library(here)
library(brms)
library(tidybayes)
```

2.1 PMA

Load in the PMA data and make an age category variable:

```
d <- read_csv(here("data/pma.csv"))

d <- d %>% filter(age>14, age<50)

age_groups <- seq(15, 45, by = 5)
d$age_group <- as.numeric(as.character(cut(d$age,
                                           breaks= c(age_groups, Inf),
                                           labels = age_groups,
                                           right = FALSE)))

d <- d %>% mutate(age_group = factor(age_group)) %>%
  mutate(age_group = fct_relevel(age_group, "35", after = 0))
```

Create a new `region_2` variable with slightly bigger regions (that match the census data):

```
d <- d %>% mutate(region_2 = case_when(region == "north"|region== "karamoja"|region == "west nile" ~ "no
                                       region == "eastern" ~ "eastern",
                                       region == "central 1"|region == "central 2"|region == "east centr
                                       region == "western"|region == "south west" ~ "western"))
```

2.2 Census

Load in the data, select the columns we want and make an age category variable:

```
dc <- haven::read_dta(here("data/uganda_census.dta"))

dc <- dc %>%
  select(regnug, perwt, age, marst, edattain) %>%
  mutate(regnug = as_factor(regnug),
         marst = as_factor(marst),
         edattain = as_factor(edattain))

dc <- dc %>% filter(age>14, age<50)


dc$age_group <- as.numeric(as.character(cut(dc$age,
                                            breaks= c(age_groups, Inf),
                                            labels = age_groups,
                                            right = FALSE)))

dc <- dc %>% mutate(age_group = factor(age_group)) %>%
  mutate(age_group = fct_relevel(age_group, "35", after = 0))
```

2.3 Recode education and marital status

The PMA and census have different education and marital status categories. Let's recode so they are the same:

```
# EDUCATION

#table(d$educattgen)
#table(dc$edattain)


d <- d %>% mutate(educ = case_when(educattgen=="never attended"~"less than primary",
                                   educattgen=="primary/middle school" ~"primary",
                                   educattgen=="secondary/post-primary"|educattgen=="tertiary/post-secon
                                   TRUE ~ "NA")) %>%
  filter(educ != "NA", marstat!="no response or missing")

dc <- dc %>% mutate(educ = case_when(edattain=="less than primary completed"~"less than primary",
                                     edattain=="primary completed"~ "primary",
                                     edattain=="secondary completed"|edattain=="university completed" ~
                                     TRUE ~ "NA")) %>%
  filter(educ !="NA")

## MARITAL STATUS

#table(d$marstat)
#table(dc$marst)

d <- d %>% mutate(marital = case_when(marstat == "never married" ~ "single/never married",
                                      marstat == "currently living with partner"| marstat == "currently
                                      marstat== "divorced or separated" ~ "divorced/separated",
                                      marstat== "widow or widower"~ "widowed",
                                      TRUE ~ "NA"))

dc <- dc %>% mutate(marital = case_when(marst == "single/never married" ~ "single/never married",
                                        marst == "married/in union" ~ "married/in union",
                                        marst== "separated/divorced/spouse absent" ~ "divorced/separate
                                        marst== "widowed"~ "widowed",
                                      TRUE ~ "NA")) %>%
  filter(marital != "NA")
```

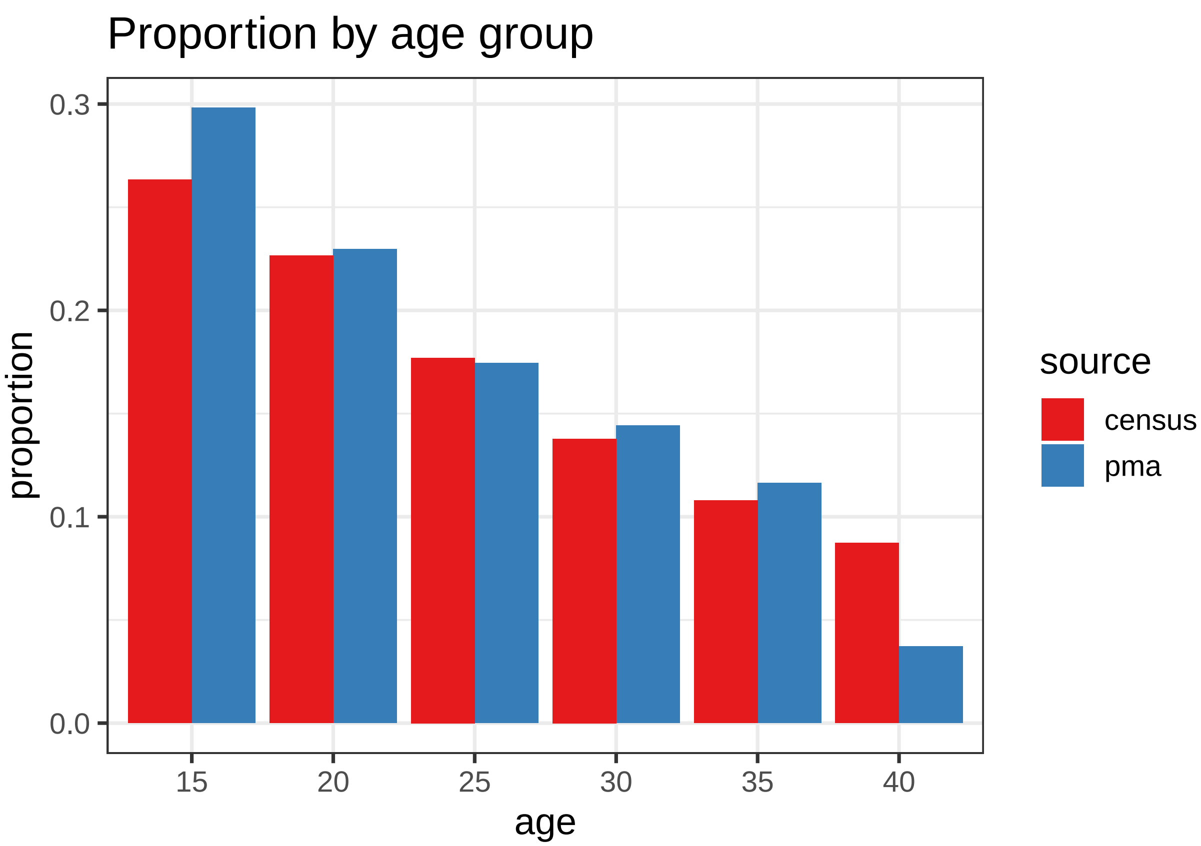
3 Plot data

Lets calculate the census population by key subgroups:

```
census_counts <- dc %>%
  group_by(regnug, marital, educ, age_group) %>%
  summarize(n = sum(perwt)) %>%
  filter(age_group!="45") %>%
  rename(region_2 = regnug)
```

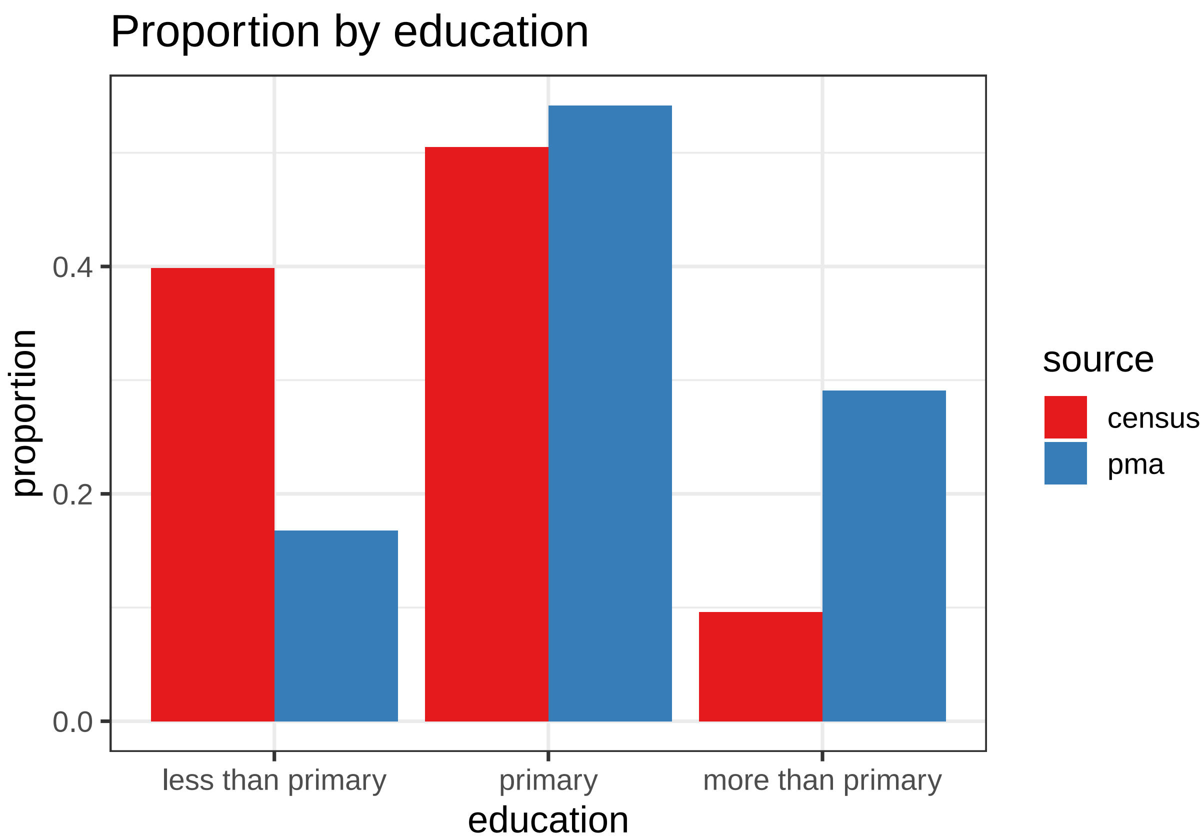
We can get an idea of differences in population distributions by plotting the PMA and census proportions by different variables:

```
d %>%
  group_by(age_group) %>%
  tally() %>%
  mutate(pma = n/sum(n)) %>%
  left_join(census_counts %>%
              group_by(age_group) %>%
              summarize(n = sum(n)) %>%
              mutate(census = n/sum(n)) %>%
              select(-n)) %>%
  mutate(age_group = as.character(age_group)) %>%
  arrange(age_group) %>%
  pivot_longer(pma:census) %>%
  ggplot(aes(age_group, value, fill = name)) + geom_bar(stat = "identity", position = 'dodge')+
  theme_bw(base_size = 14) +
  labs( x = "age", y = "proportion", title = "Proportion by age group")+
  scale_fill_brewer(palette = "Set1", name = "source")
```

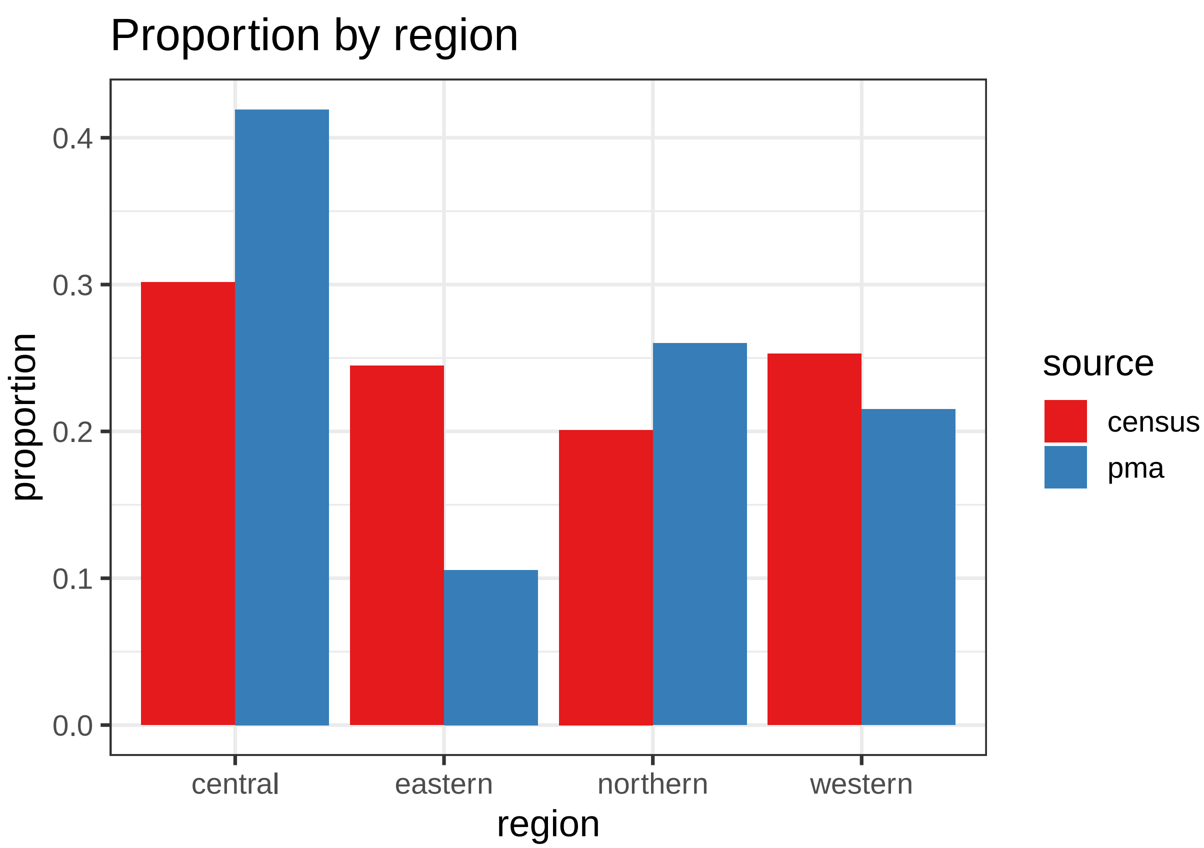



```
d %>%
  group_by(educ) %>%
  tally() %>%
  mutate(pma = n/sum(n)) %>%
  left_join(census_counts %>%
              group_by(educ) %>%
              summarize(n = sum(n)) %>%
              mutate(census = n/sum(n)) %>%
              select(-n)) %>%
```

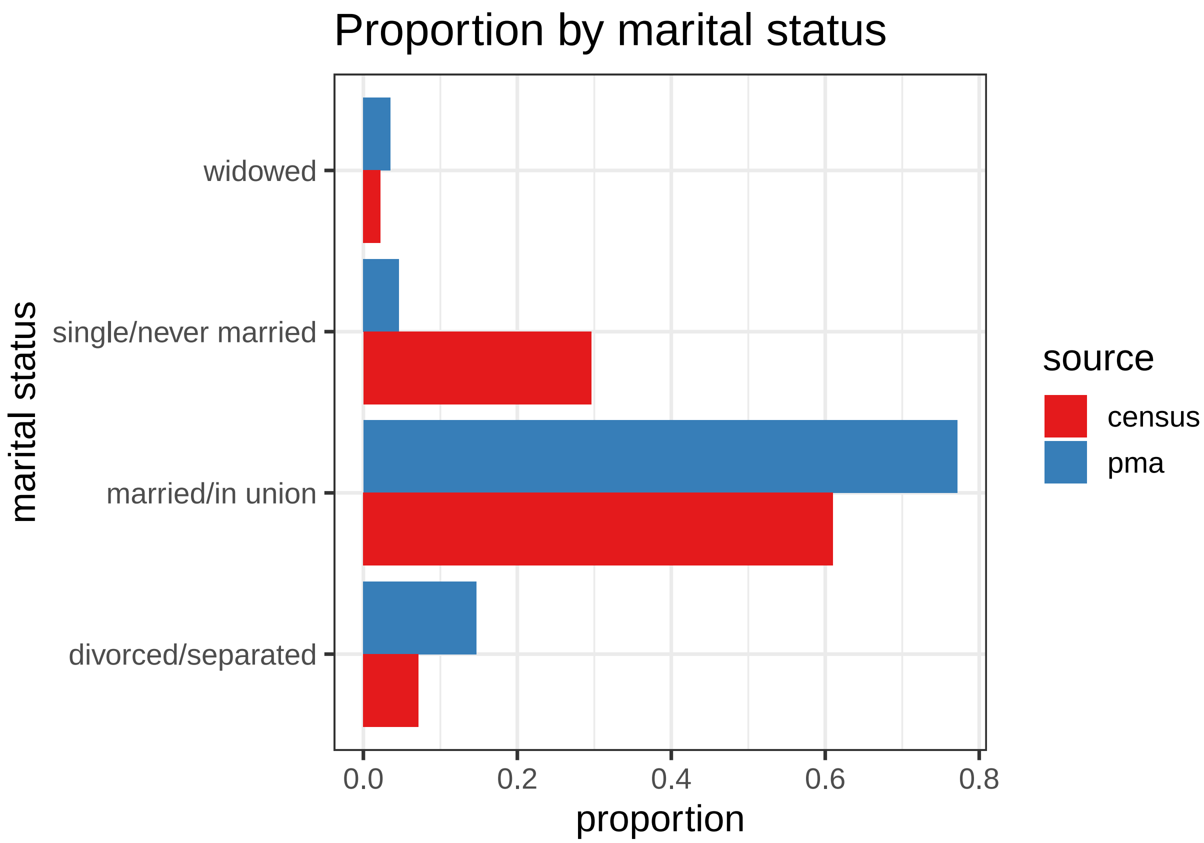
```
  mutate(educ = fct_relevel(educ, "more than primary", after = 2)) %>%
  pivot_longer(pma:census) %>%
  ggplot(aes(educ, value, fill = name)) + geom_bar(stat = "identity", position = 'dodge')+
  theme_bw(base_size = 14) +
  labs( x = "education", y = "proportion", title = "Proportion by education")+
  scale_fill_brewer(palette = "Set1", name = "source")
```




```
d %>%
  group_by(region_2) %>%
  tally() %>%
  mutate(pma = n/sum(n)) %>%
  left_join(census_counts %>%
              group_by(region_2) %>%
              summarize(n = sum(n)) %>%
              mutate(census = n/sum(n)) %>%
              select(-n)) %>%
  pivot_longer(pma:census) %>%
  ggplot(aes(region_2, value, fill = name)) + geom_bar(stat = "identity", position = 'dodge')+
  theme_bw(base_size = 14) +
  labs( x = "region", y = "proportion", title = "Proportion by region")+
  scale_fill_brewer(palette = "Set1", name = "source")
```




```
d %>%
  group_by(marital) %>%
  tally() %>%
  mutate(pma = n/sum(n)) %>%
  left_join(census_counts %>%
              group_by(marital) %>%
              summarize(n = sum(n)) %>%
              mutate(census = n/sum(n)) %>%
              select(-n)) %>%
  pivot_longer(pma:census) %>%
  ggplot(aes(marital, value, fill = name)) + geom_bar(stat = "identity", position = 'dodge')+
  theme_bw(base_size = 14) +
  labs( x = "marital status", y = "proportion", title = "Proportion by marital status")+
  scale_fill_brewer(palette = "Set1", name = "source")+coord_flip()
```



4 Multilevel regression

Let's run a logistic regression of whether or not an individual reported ever having an abortion with covariates education, age group, and region modelled hierarchically

4.1 Run the model

```
mod <- brm(abortion ~ (1|region_2)+educ+ marital +age_group, data = d, family = "bernoulli")
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## clang -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Library/Frameworks/R.frame
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
```

```
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: fa
## #include <complex>
##          ^~~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
##
## SAMPLING FOR MODEL '3a34a904dd27e5fa1013c2defb8b97c2' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.0002 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 6.76701 seconds (Warm-up)
## Chain 1:                5.04187 seconds (Sampling)
## Chain 1:                11.8089 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '3a34a904dd27e5fa1013c2defb8b97c2' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.000162 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 1.62 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 5.62723 seconds (Warm-up)
```

```
## Chain 2:                5.23309 seconds (Sampling)
## Chain 2:                10.8603 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '3a34a904dd27e5fa1013c2defb8b97c2' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.000255 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 2.55 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 5.79643 seconds (Warm-up)
## Chain 3:                6.36309 seconds (Sampling)
## Chain 3:                12.1595 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '3a34a904dd27e5fa1013c2defb8b97c2' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.000134 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 1.34 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 4.88351 seconds (Warm-up)
## Chain 4:                5.57253 seconds (Sampling)
## Chain 4:                10.456 seconds (Total)
## Chain 4:
```

```r
summary(mod)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: abortion ~ (1 | region_2) + educ + marital + age_group
##    Data: d (Number of observations: 2774)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~region_2 (Number of levels: 4)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.84      0.57     0.26     2.64 1.05       84       37
##
## Population-Level Effects:
##                           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                    -2.42      0.60    -3.95    -1.29 1.05      110
## educmorethanprimary           0.78      0.25     0.31     1.29 1.00     1723
## educprimary                   0.70      0.22     0.28     1.15 1.01     2190
## maritalmarriedDinunion       -0.47      0.15    -0.75    -0.18 1.00     2759
## maritalsingleDnevermarried   -1.17      0.37    -1.93    -0.47 1.00     3182
## maritalwidowed               -0.95      0.44    -1.78    -0.11 1.02      237
## age_group15                   0.01      0.22    -0.42     0.43 1.00     1477
## age_group20                   0.12      0.22    -0.31     0.57 1.01     1616
## age_group25                   0.34      0.23    -0.09     0.78 1.01     1537
## age_group30                   0.09      0.25    -0.40     0.58 1.01     1808
## age_group40                   0.03      0.37    -0.73     0.73 1.00     2008
##                           Tail_ESS
## Intercept                       23
## educmorethanprimary           2187
## educprimary                   2263
## maritalmarriedDinunion        2704
## maritalsingleDnevermarried    2667
## maritalwidowed                 333
## age_group15                   2052
## age_group20                   2185
## age_group25                   2185
## age_group30                   2010
## age_group40                   2452
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
# optional: cf to lme4
#summary(lme4::glmer(abortion ~ (1|region_2)+marital+educ+ age_group, data = d, family = "binomial"))
```

4.2 Get estimated proportions of abortion incidence

Now we need to get the estimated proportions of abortion incidence by each subpopulation of interest. (i.e. by region, marital status, education and age group). First, initiate a tibble with all possible combinations:

```
regions <- unique(d$region_2)
marital_groups <- unique(d$marital)
educ_groups  <- unique(d$educ)

pred_df <- tibble(region_2 = NA, marital = NA, educ = NA, age_group = NA) %>%
  tidyr::complete(region_2 = regions,
                  marital = marital_groups,
                  educ = educ_groups,
                  age_group = age_groups) %>%
  mutate_all(.funs = funs(as_factor(.))) %>%
  drop_na() %>%
  mutate(age_group = fct_relevel(age_group, "35", after = 0)) %>%
  filter(age_group != "45")
```

We can use the `fitted_draws` function to obtain posterior samples of the estimated proportion of women reporting an abortion by each group:

```
pred_probs_draws <- mod %>%
  fitted_draws(pred_df  %>%
                  select(region_2:age_group) %>%
                  mutate(age_group = factor(age_group, levels = age_groups[-length(age_groups)])))
```

5 Post-stratify the predicted proportions

For the post-stratfication, we need a data frame that tells us how many women in the census are in each subgroup of interest. Let's join the census counts to the `pred_probs_draws` table above, replacing any NAs with 0's:

```
pred_probs_draws <- pred_probs_draws %>%
  left_join(census_counts) %>%
  replace_na(replace = list(n = 0))
```

For each group and draw, we can calculate the estimated number of women reporting an abortion:

```
pred_probs_draws <- pred_probs_draws %>%
  mutate(n_abo = n*.value)
```

We can now use this as a basis of getting estimates by any group of interest. For example, the national estimate and lower/upper bounds is

```
pred_probs_draws %>%
  group_by(.draw) %>%
  summarize(prop_abo = sum(n_abo)/sum(n)) %>%
  ungroup() %>%
  summarize(prop_abo_group = median(prop_abo),
            lower = quantile(prop_abo, 0.1),
            upper = quantile(prop_abo, 0.9))
```

```
## # A tibble: 1 x 3
##   prop_abo_group  lower upper
##            <dbl>  <dbl> <dbl>
## 1         0.0900 0.0802 0.102
```

By age group:

```
prop_by_age <- pred_probs_draws %>%
  group_by(.draw, age_group) %>%
  summarize(prop_abo = sum(n_abo)/sum(n)) %>%
  group_by(age_group) %>%
  summarize(prop_abo_group = median(prop_abo),
            lower = quantile(prop_abo, 0.1),
            upper = quantile(prop_abo, 0.9))

prop_by_age
```

```
## # A tibble: 6 x 4
##   age_group prop_abo_group  lower  upper
##   <fct>              <dbl>  <dbl>  <dbl>
## 1 15                0.0640 0.0483 0.0842
## 2 20                0.0953 0.0802 0.113
## 3 25                0.123  0.105  0.143
## 4 30                0.0931 0.0761 0.113
## 5 35                0.0841 0.0676 0.104
## 6 40                0.0849 0.0549 0.118
```

By marital status:

```
prop_by_marital <- pred_probs_draws %>%
  group_by(.draw, marital) %>%
  summarize(prop_abo = sum(n_abo)/sum(n)) %>%
  group_by(marital) %>%
  summarize(prop_abo_group = median(prop_abo),
            lower = quantile(prop_abo, 0.1),
            upper = quantile(prop_abo, 0.9))

prop_by_marital
```

```
## # A tibble: 4 x 4
##   marital              prop_abo_group  lower  upper
##   <chr>                         <dbl>  <dbl>  <dbl>
## 1 divorced/separated           0.147  0.125  0.171
## 2 married/in union             0.0995 0.0897 0.111
## 3 single/never married         0.0578 0.0373 0.0853
## 4 widowed                      0.0544 0.0305 0.0867
```

By education:

```
prop_by_educ <- pred_probs_draws %>%
  group_by(.draw, educ) %>%
  summarize(prop_abo = sum(n_abo)/sum(n)) %>%
  group_by(educ) %>%
  summarize(prop_abo_group = median(prop_abo),
            lower = quantile(prop_abo, 0.1),
            upper = quantile(prop_abo, 0.9))

prop_by_educ
```

```
## # A tibble: 3 x 4
##   educ               prop_abo_group  lower  upper
##   <chr>                       <dbl>  <dbl>  <dbl>
## 1 less than primary          0.0578 0.0442 0.0741
## 2 more than primary          0.122  0.104  0.143
## 3 primary                    0.109  0.0960 0.124
```

By region:

```
prop_by_region <- pred_probs_draws %>%
  group_by(.draw, region_2) %>%
  summarize(prop_abo = sum(n_abo)/sum(n)) %>%
  group_by(region_2) %>%
  summarize(prop_abo_group = median(prop_abo),
            lower = quantile(prop_abo, 0.1),
            upper = quantile(prop_abo, 0.9))

prop_by_region
```

```
## # A tibble: 4 x 4
##   region_2 prop_abo_group  lower  upper
##   <fct>             <dbl>  <dbl>  <dbl>
## 1 central          0.118  0.104  0.134
## 2 eastern          0.120  0.0983 0.145
## 3 northern         0.0625 0.0520 0.0749
## 4 western          0.0482 0.0385 0.0601
```

6 Plotting and comparing estimates

We can compare the MRP estimates with raw estimates from the PMA survey and also estimates using normal post-stratification

6.1 Calculating raw estimates and post-stratified estimates

Calculate the raw estimates from the survey:

```
pma_cells <- d %>%
  group_by(region_2, educ, marital, age_group) %>%
  summarize(
            n_sample = n(), n_abo_sample = sum(abortion),
            prop_abo = n_abo_sample/n_sample)
```

We can then combine these with census data to get post-stratified counts:

```
pma_cells <- pma_cells %>%
  left_join(census_counts) %>%
  mutate(n_abo = prop_abo*n) %>%
  replace_na(replace = list(n = 0,n_abo = 0)) %>%
  ungroup()
```

6.2 Join all estimates together

Calculate raw and postratified estimates by different subpopulations and join them to the MRP estimates.
By age group:

```
prop_by_age <- prop_by_age %>%
  mutate(type = "mrp") %>%
  rename(point = prop_abo_group) %>%
  bind_rows(pma_cells %>% group_by(age_group) %>%summarize(poststrat = sum(n_abo)/sum(n)) %>%
  left_join(pma_cells %>% group_by(age_group) %>%summarize(raw = sum(n_abo_sample)/sum(n_sample))) %>%
  pivot_longer(-age_group, names_to = "type", values_to = "point")) %>%
  arrange(age_group)
```

By education:

```
prop_by_educ <- prop_by_educ %>%
  mutate(type = "mrp") %>%
  rename(point = prop_abo_group) %>%
  bind_rows(pma_cells %>% group_by(educ) %>%summarize(poststrat = sum(n_abo)/sum(n)) %>%
  left_join(pma_cells %>% group_by(educ) %>%summarize(raw = sum(n_abo_sample)/sum(n_sample))) %>%
  pivot_longer(-educ, names_to = "type", values_to = "point")) %>%
  arrange(educ)
```

By marital status:

```
prop_by_marital <- prop_by_marital %>%
  mutate(type = "mrp") %>%
  rename(point = prop_abo_group) %>%
  bind_rows(pma_cells %>% group_by(marital) %>%summarize(poststrat = sum(n_abo)/sum(n)) %>%
  left_join(pma_cells %>% group_by(marital) %>%summarize(raw = sum(n_abo_sample)/sum(n_sample))) %>%
  pivot_longer(-marital, names_to = "type", values_to = "point")) %>%
  arrange(marital)
```

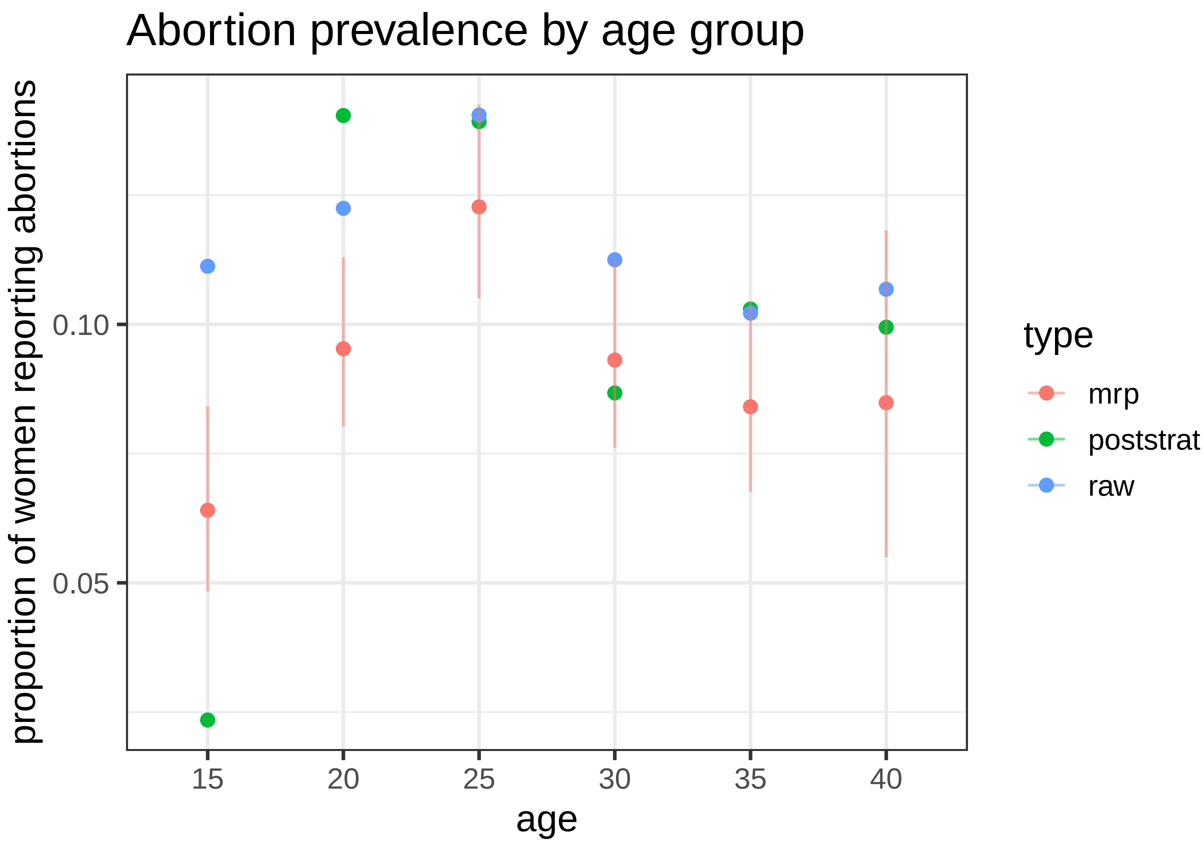
By region:

```
prop_by_region <- prop_by_region %>%
  mutate(type = "mrp") %>%
  rename(point = prop_abo_group) %>%
  bind_rows(pma_cells %>% group_by(region_2) %>%summarize(poststrat = sum(n_abo)/sum(n)) %>%
  left_join(pma_cells %>% group_by(region_2) %>%summarize(raw = sum(n_abo_sample)/sum(n_sample))) %>%
  pivot_longer(-region_2, names_to = "type", values_to = "point")) %>%
  arrange(region_2)
```

6.3 Plot!

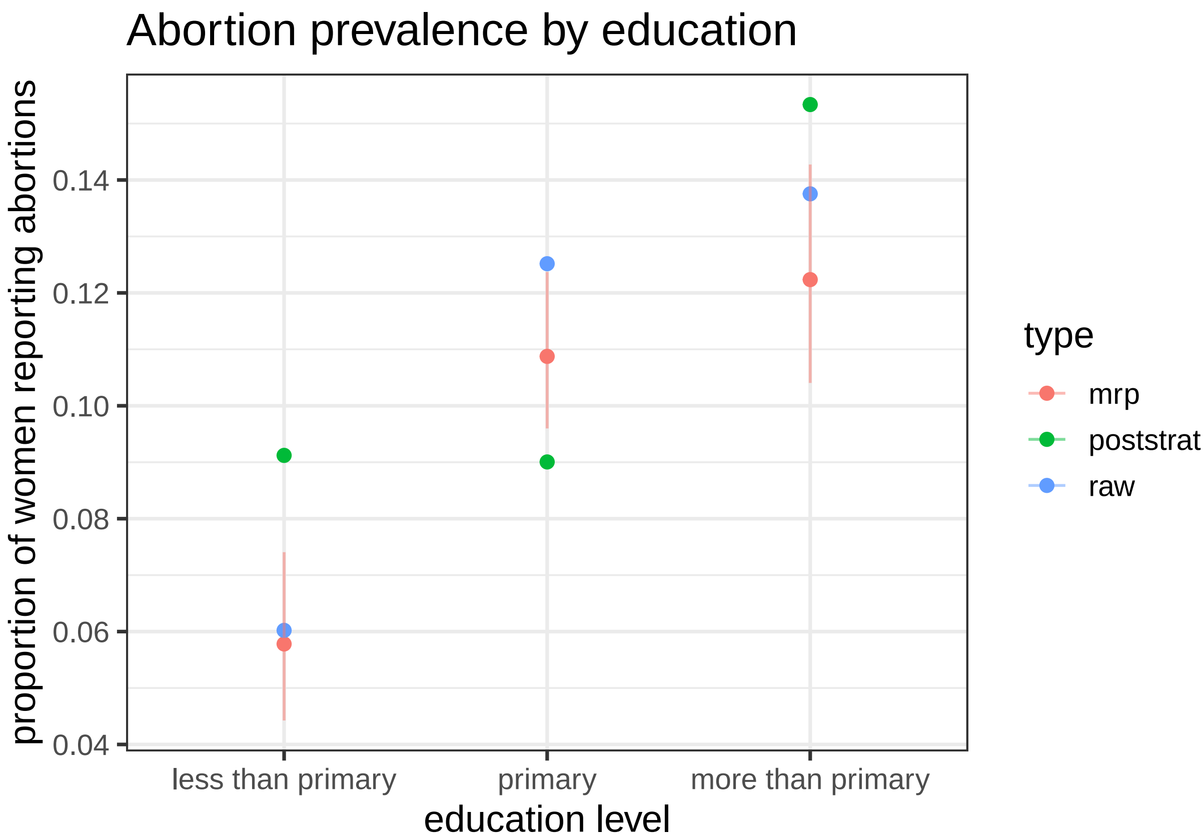
By age group:

```
prop_by_age %>%
  ggplot(aes(age_group, point, color = type)) + geom_point(size = 2) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = NA, alpha = 0.5) +
  theme_bw(base_size = 14) +
  labs(title = "Abortion prevalence by age group", x = "age", y = "proportion of women reporting aborti
```



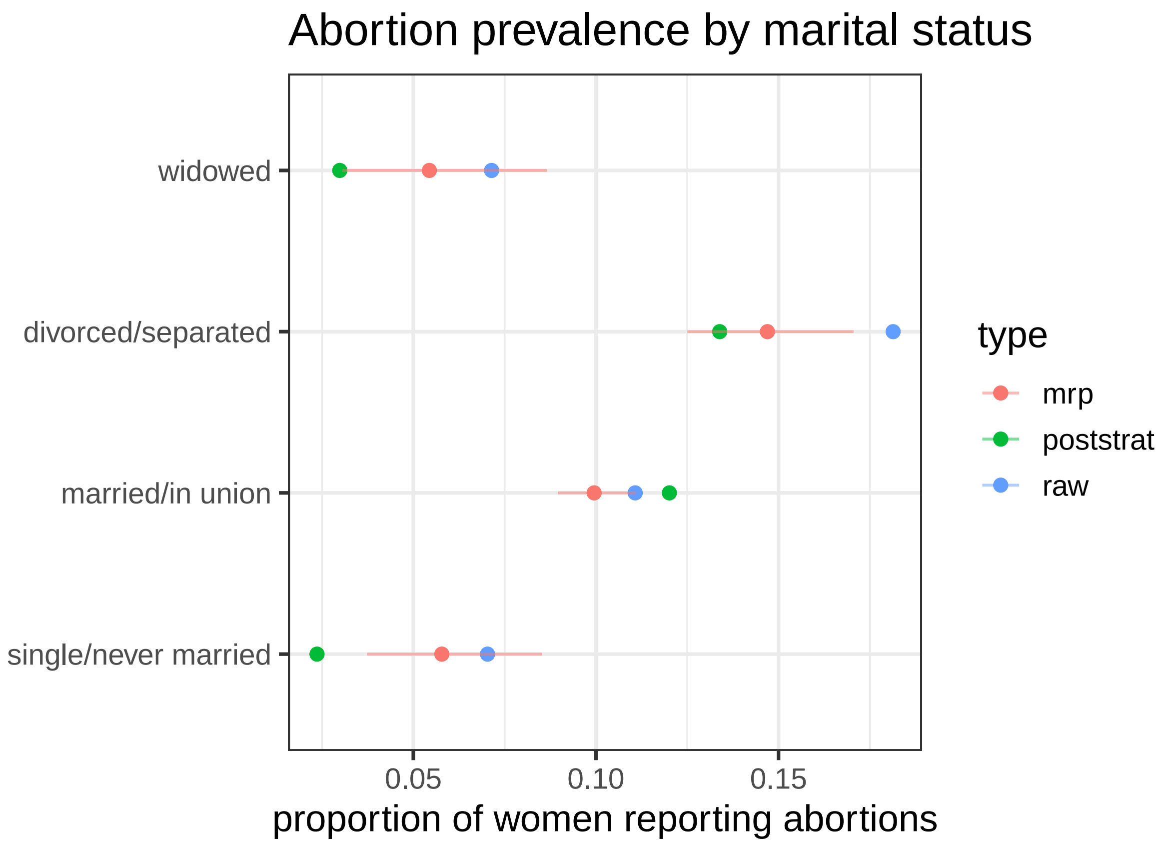
By education:

```
prop_by_educ %>%
  mutate(educ = fct_relevel(educ, "more than primary", after = 2)) %>%
  ggplot(aes(educ, point, color = type)) + geom_point(size = 2) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = NA, alpha = 0.5) +
  theme_bw(base_size = 14) +
  labs(title = "Abortion prevalence by education", x = "education level", y = "proportion of women repo
```



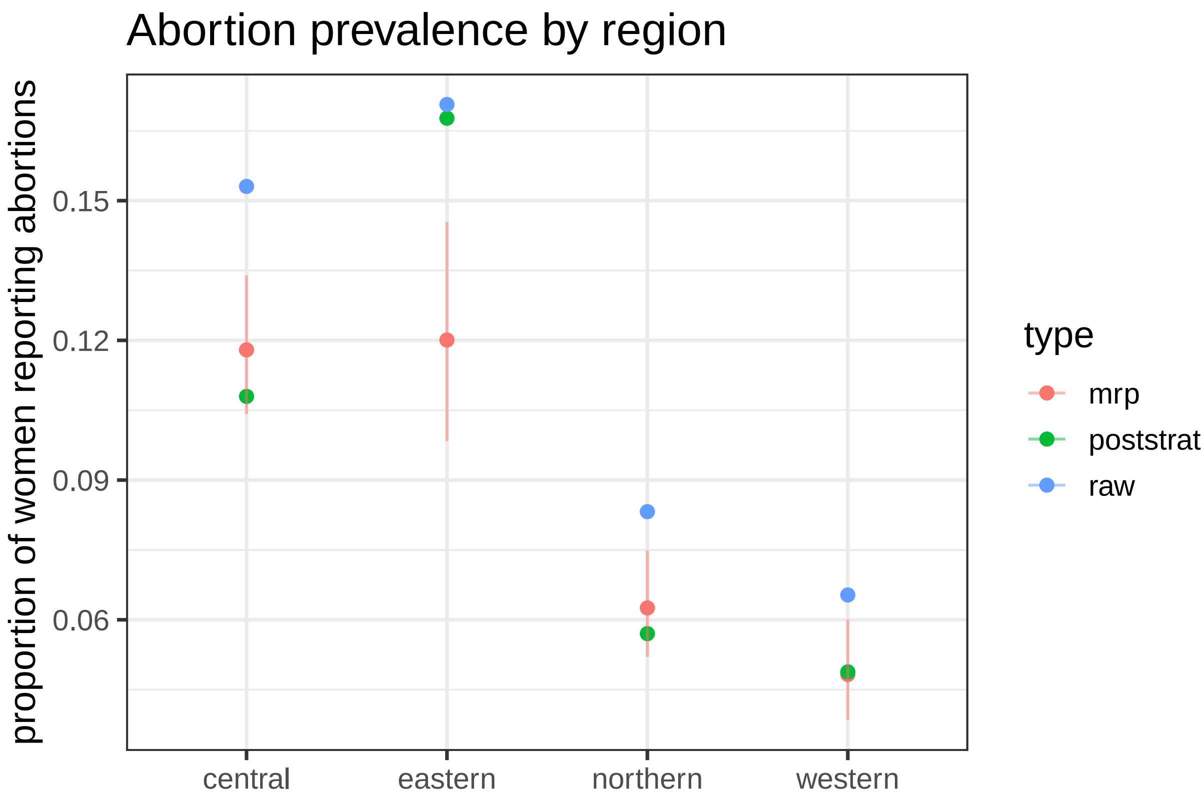
By marital status

```
prop_by_marital %>%
  mutate(marital = factor(marital, c("single/never married", "married/in union", "divorced/separated",
  ggplot(aes(marital, point, color = type)) + geom_point(size = 2) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = NA, alpha = 0.5) +
  theme_bw(base_size = 14) +
  labs(title = "Abortion prevalence by marital status", x = "", y = "proportion of women reporting abor
  coord_flip()
```



By region

```
prop_by_region %>%
  ggplot(aes(region_2, point, color = type)) + geom_point(size = 2) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = NA, alpha = 0.5) +
  theme_bw(base_size = 14) +
  labs(title = "Abortion prevalence by region", x = "", y = "proportion of women reporting abortions")
```


Abortion prevalence by region
proportion of women reporting abortions
0.15
0.12
0.09
0.06
central
eastern
northern
western
type
mrp
poststrat
raw