Chapter 16 Fertility in Cambodia

John Bryant and Junni L. Zhang

2018-09-10

Setup

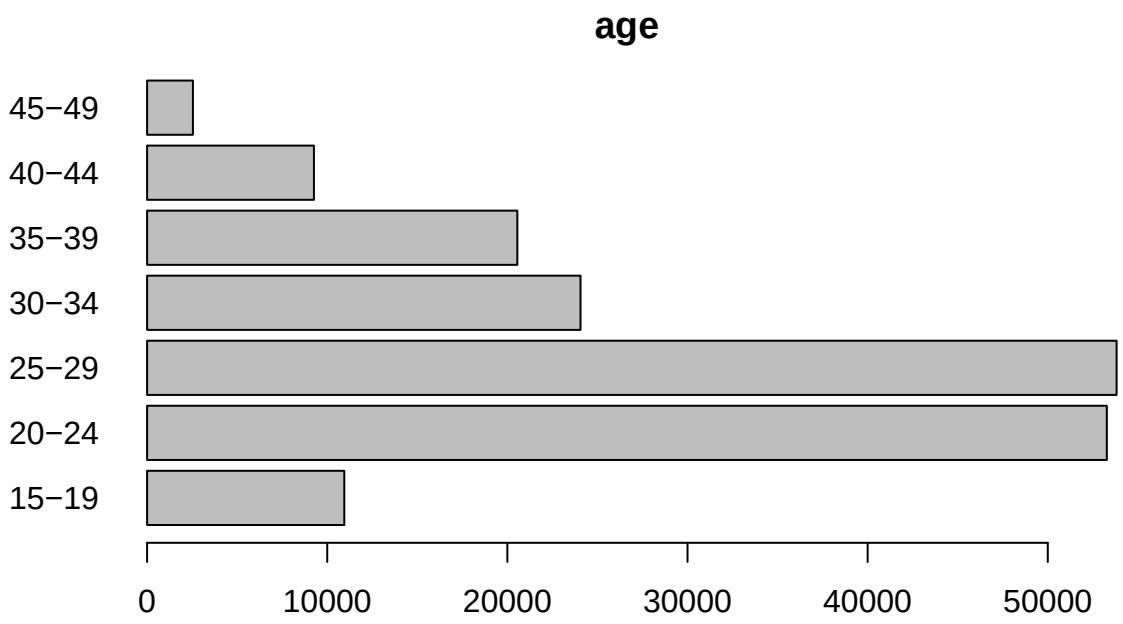
```
library(methods)
library(bdefdata)
library(demest)
library(dplyr)
library(tidyverse)
library(latticeExtra)
```

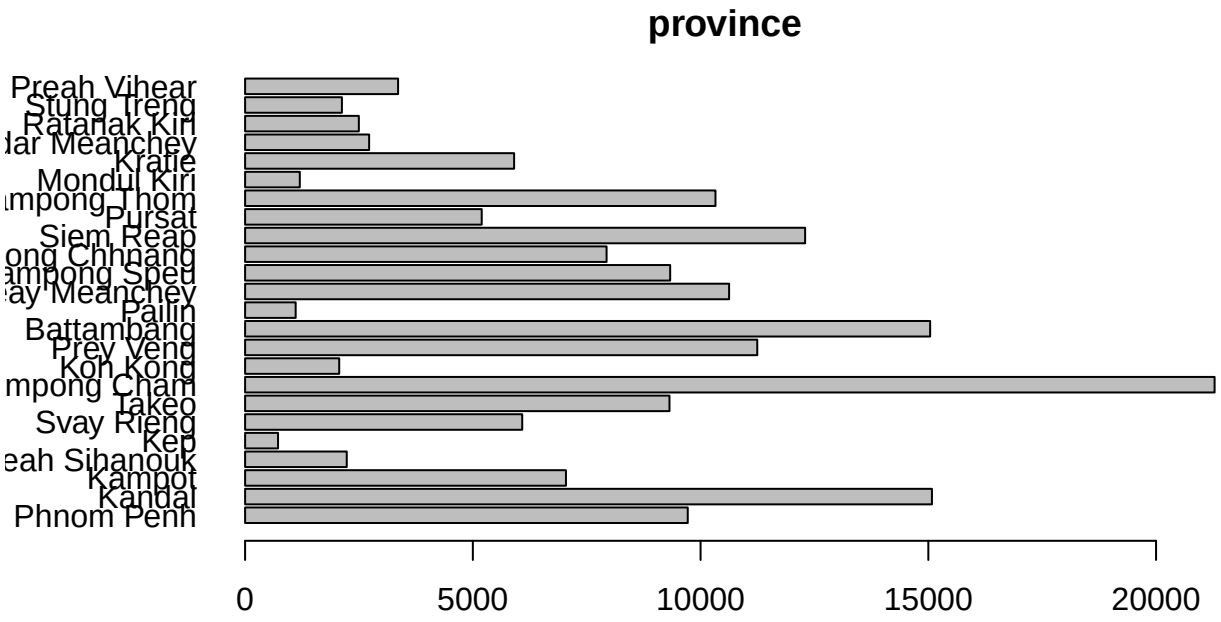
Get data

```
births_census <- bdefdata::cambodia_births_census %>%
  Counts()
summary(births_census)
```

```
##
## name:     age       province
## length:   7         24
## dimtype:  age       state
## dimscale: Intervals Categories
## first:    15-19     Phnom Penh
## last:     45-49     Preah Vihear
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     5.0   220.8   536.5  1038.5  1387.2  6370.0
```

```
plot(births_census)
```





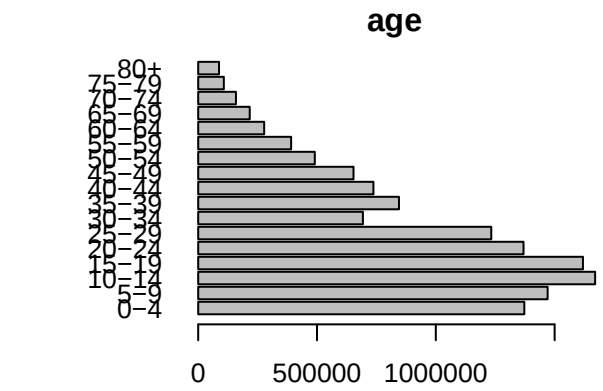

```
births_dhs <- bdefdata::cambodia_births_dhs
births_dhs
```

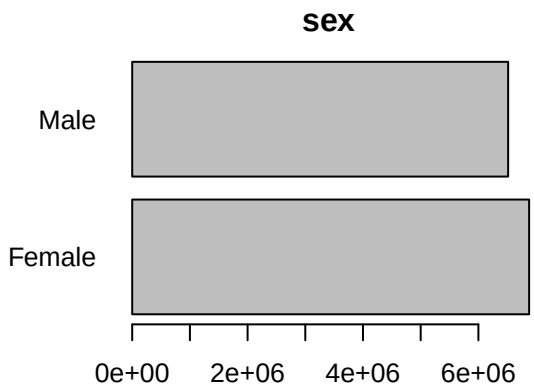
```
##     age      births       se
## 1 15-19  26575.1446 2795.454
## 2 20-24 101737.0683 5341.732
## 3 25-29 115264.6218 6292.452
## 4 30-34  55207.9321 4288.545
## 5 35-39  32737.9213 3302.006
## 6 40-44  12103.1726 1854.812
## 7 45-49    198.6996  128.597
```

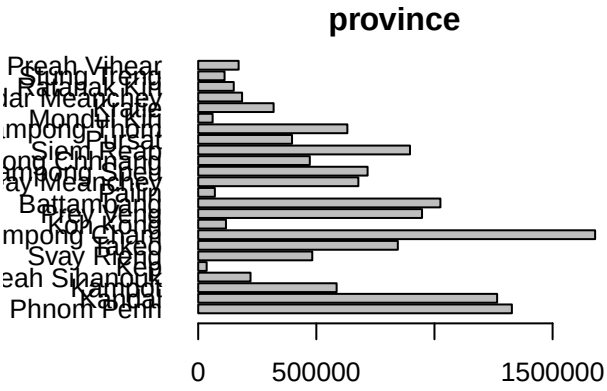
```
population <- bdefdata::cambodia_population %>%
  Counts()
summary(population)
```

```
##
## name:     age       sex    province
## length:   17        2      24
## dimtype:  age       sex    state
## dimscale: Intervals Sexes  Categories
## first:    0-4       Female Phnom Penh
## last:     80+       Male   Preah Vihear
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      62    2713    7962   16416   23337  116474
```

```
plot(population)
```








```
poverty <- bdefdata::cambodia_poverty
poverty
```

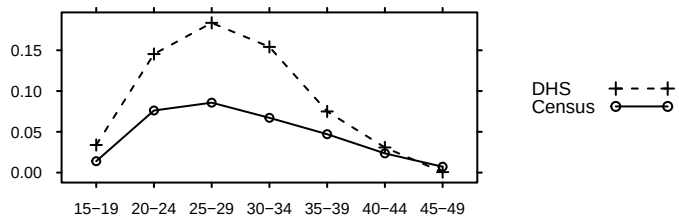
```
##            province poverty
## 1        Phnom Penh     0.2
## 2            Kandal    17.6
## 3            Kampot    20.5
## 4    Preah Sihanouk    21.1
## 5               Kep    22.8
## 6        Svay Rieng    23.6
```

```
## 7             Takeo    25.2
## 8       Kampong Cham    25.8
## 9           Koh Kong    26.7
## 10         Prey Veng    27.3
## 11        Battambang    29.9
## 12            Pailin    31.0
## 13 Banteay Meanchey    31.3
## 14      Kampong Speu    32.2
## 15  Kampong Chhnang    32.3
## 16         Siem Reap    32.4
## 17            Pursat    34.1
## 18      Kampong Thom    34.4
## 19       Mondul Kiri    38.0
## 20            Kratie    38.6
## 21   Otdar Meanchey    38.6
## 22      Ratanak Kiri    41.5
## 23       Stung Treng    42.4
## 24      Preah Vihear    43.1
```

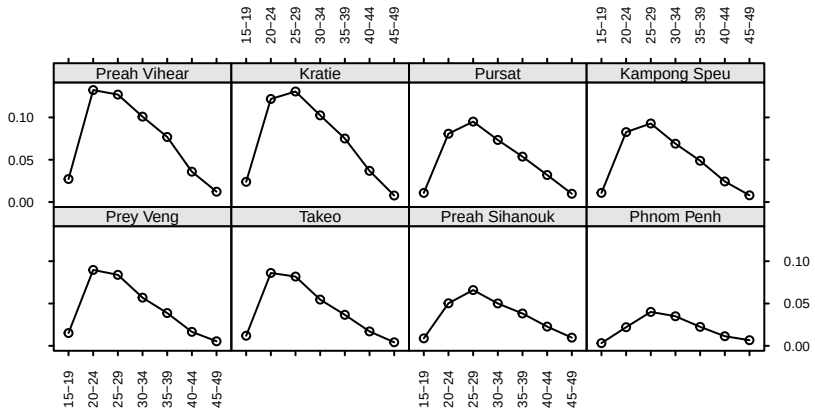
Plot direct estimates

```
#Direct estimates of age-specific fertility rates at the nation level,
#based on the 2008 Census and 2010 Demographic and Health Survey.
births_dhs_nation <- births_dhs %>%
    xtabs(births ~ age, data = .) %>%
    Counts()
births_census_nation <- births_census %>%
    collapseDimension(margin = "age")
exposure_nation <- population %>%
    subarray(sex == "Female" & age > 15 & age < 50) %>%
    collapseDimension(margin = "age")

rates <- dbind(DHS = births_dhs_nation / exposure_nation,
               Census = births_census_nation / exposure_nation,
               along = "variant") %>%
    as.data.frame(direction = "long")

lty <- c("dashed", "solid")
pch <- c(3, 1)
xyplot(value ~ age,
       data = rates,
       groups = variant,
       col = "black",
       type = "b",
       pch = pch,
       lty = lty,
       xlab = "",
       ylab = "",
       scales = list(tck = 0.3),
       par.settings = list(fontsize = list(text = 7, points = 5),
                           strip.background = list(col = "grey90")),
       key = list(text = list(levels(rates$variant)),
                  lines = list(col = "black", lty = lty, type = "b", pch = pch, cex = 0.9),
```

```
                space = "right",
                divide = 2))
```




```
#Direct estimates of age-specific fertility rates for eight selected
#provinces, based on the 2008 Census.
births_province <- births_census %>%
    collapseDimension(margin = c("age", "province"))
exposure_province <- population %>%
    subarray(sex == "Female" & age > 15 & age < 50)

levels_province <- dimnames(exposure_province)$province
n <- length(levels_province)

sample_provinces <- levels_province[seq(from = 1, to = n, length.out = n / 3)]

rates <- (births_province / exposure_province) %>%
    subarray(province %in% sample_provinces) %>%
    as.data.frame(direction = "long") %>%
    mutate(province = factor(province, levels = rev(sample_provinces)))

xyplot(value ~ age | province,
       data = rates,
       col = "black",
       type = "b",
       pch = 1,
       xlab = "",
       ylab = "",
       as.table = TRUE,
       layout = c(NA, 2),
       scales = list(tck = 0.3,
                     x = list(rot = 90)),
       par.settings = list(fontsize = list(text = 6, points = 5),
                           strip.background = list(col = "grey90")))
```



Fit the baseline model

```
filename_base <- "cambodia_base.est"
if(!file.exists(filename_base)) {
    #Get datasets
    births_dhs_data <- CountsOne(births_dhs$births,
                                 labels = births_dhs$age,
                                 name = "age") %>%
        toInteger(force = TRUE)

    datasets <- list(births_census = births_census,
                     births_dhs = births_dhs_data)

    #Set up system models
    model <- Model(y ~ Poisson(mean ~ age + province),
                   age ~ DLM(damp = NULL),
                   province ~ Exch(covariates = Covariates(mean ~ poverty, data = poverty)),
                   jump = 0.05)

    #Set up data models
    #Data model for census
    model_census <- Model(births_census ~ Poisson(mean ~ age + province),
                          age ~ DLM(trend = NULL,
                                    damp = NULL),
                          province ~ Exch(error = Error(scale = HalfT(scale = 0.05))),
                          jump = 0.07)

    #Data model for DHS
    mean_dhs <- ValuesOne(rep(1, times = nrow(births_dhs)),
                          labels = births_dhs$age,
                          name = "age")
    se_dhs <- ValuesOne(births_dhs$se,
                        labels = births_dhs$age,
                        name = "age")

    model_dhs <- Model(births_dhs ~ NormalFixed(mean = mean_dhs, sd = se_dhs))
```

```
    data_models <- list(model_dhs, model_census)

    #Estimation
    y <- toInteger(2 * datasets$births_census) # starting values for calculations

    n_burnin <- 200000
    n_sim <- 250000
    n_chain <-4
    n_thin <- 500

    set.seed(0)

    estimateCounts(model = model,
                   y = y,
                   exposure = exposure_province,
                   dataModels = data_models,
                   datasets = datasets,
                   filename = filename_base,
                   nBurnin = n_burnin,
                   nSim = n_sim,
                   nChain = n_chain,
                   nThin = n_thin)

}
fetchSummary(filename_base)
```

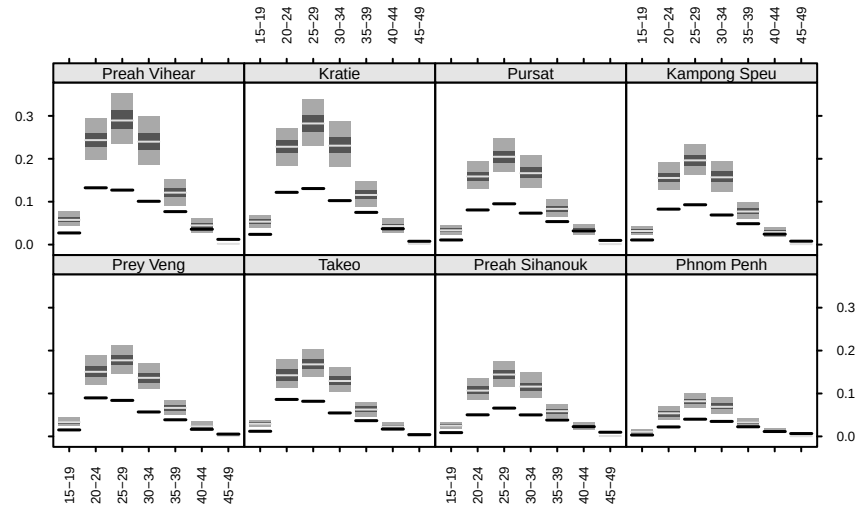
```
## ----------------------------------------------------
## model:
## y ~ Poisson(mean ~ age + province),
## age ~ DLM(damp = NULL),
## province ~ Exch(covariates = Covariates(mean ~ poverty, data = poverty)),
## 0.05
## dimensions: age, province
## ----------------------------------------------------
## y:
## dimensions: age, province
## n cells: 168
## ----------------------------------------------------
## Data models:
## *births_census*
## births_census ~ Poisson(mean ~ age + province),
## age ~ DLM(trend = NULL, damp = NULL),
## province ~ Exch(error = Error(scale = HalfT(scale = 0.05))),
## 0.07
## dimensions: age, province
## *births_dhs*
## births_dhs ~ NormalFixed(mean = mean_dhs, sd = se_dhs)
## dimensions: age
## ----------------------------------------------------
## Datasets:
## *births_census*
## Object of class "Counts"
## dimensions: age, province
## n cells: 168,  n missing: 0, integers: TRUE, n zeros: 0, median: 536.5
```

```
## *births_dhs*
## Object of class "Counts"
## dimensions: age
## n cells: 7,  n missing: 0, integers: TRUE, n zeros: 0, median: 32738
## ---------------------------------------------------
## MCMC statistics:
## nBurnin: 200000,  nSim: 200000,  nChain: 4,  nThin: 500,  nIteration: 1600
##
## Metropolis-Hastings updates:
##                                          jump acceptance autocorr
## model.likelihood.rate                    0.05      0.366    0.540
## dataModels.births_census.likelihood.rate 0.07      0.398    0.518
##
## parameters:
##                                                    Rhat     2.5%    50%
## model.likelihood.rate                              1.05 0.000885 0.0797
## model.prior.mean                                   1.06    -7.02  -2.54
## model.prior.sd                                     1.04   0.0055 0.0603
## model.hyper.age.scaleLevel                         1.01   0.0421  0.505
## model.hyper.age.scaleTrend                         1.00     0.34  0.804
## model.hyper.age.scaleError                         1.00   0.0449  0.514
## model.hyper.province.coef                          1.00    0.412  0.538
## model.hyper.province.scaleError                    1.01   0.0909  0.157
## y                                                  1.05        5   1063
## dataModels.births_census.likelihood.rate           1.07     0.34  0.546
## dataModels.births_census.prior.mean                1.05   -0.989 -0.621
## dataModels.births_census.prior.sd                  1.06    0.019  0.129
## dataModels.births_census.hyper.age.scaleLevel      1.00    0.207  0.801
## dataModels.births_census.hyper.age.scaleError      1.00   0.0503  0.434
## dataModels.births_census.hyper.province.scaleError 1.00   0.0025 0.0346
##                                                    97.5% length
## model.likelihood.rate                              0.271    168
## model.prior.mean                                   -1.29    168
## model.prior.sd                                     0.158      1
## model.hyper.age.scaleLevel                          1.88      1
## model.hyper.age.scaleTrend                          1.44      1
## model.hyper.age.scaleError                          1.53      1
## model.hyper.province.coef                           0.62      1
## model.hyper.province.scaleError                    0.217      1
## y                                                   9248    168
## dataModels.births_census.likelihood.rate            8.37    168
## dataModels.births_census.prior.mean                 2.05    168
## dataModels.births_census.prior.sd                  0.165      1
## dataModels.births_census.hyper.age.scaleLevel       1.37      1
## dataModels.births_census.hyper.age.scaleError       1.17      1
## dataModels.births_census.hyper.province.scaleError 0.0717      1
## ---------------------------------------------------
```

Results of the baseline model

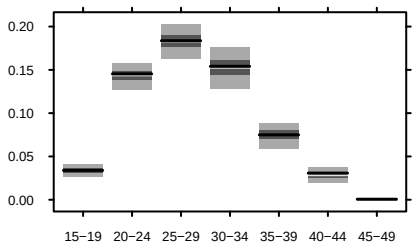
```
#Estimates of age-specific fertility rates for eight selected provinces
rates <- fetch(filename_base,
               where = c("model", "likelihood", "rate")) %>%
```

```
    slab(dimension = "province", elements = rev(sample_provinces))

births_census <- fetch(filename_base,
                       where = c("datasets", "births_census"))
exposure <- fetch(filename_base,
                  where = "exposure")
rates_direct <- (births_census / exposure_province) %>%
    subarray(province %in% sample_provinces) %>%
    slab(dimension = "province", elements = rev(sample_provinces))

dplot(~ age | province,
      data = rates,
      col = "grey50",
      xlab = "",
      ylab = "",
      as.table = TRUE,
      scales = list(tck = 0.3,
                    x = list(rot = 90)),
      layout = c(NA, 2),
      par.settings = list(fontsize = list(text = 6),
                          strip.background = list(col = "grey90")),
      overlay = list(values = rates_direct,
                     col = "black",
                     lwd = 1.5))
```

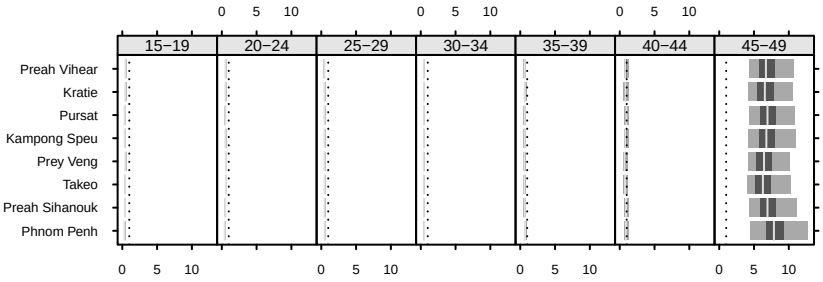



```
#Estimates of age-specific fertility rates for the whole country
rates <- fetch(filename_base,
               where = c("model", "likelihood", "rate"))

births_dhs <- fetch(filename_base,
                    where = c("datasets", "births_dhs"))
exposure <- fetch(filename_base,
                  where = "exposure")
rates_direct <- births_dhs / exposure_province
```

```
dplot(~ age,
      data = rates,
      weights = exposure,
      col = "grey50",
      xlab = "",
      ylab = "",
      scales = list(tck = 0.3),
      par.settings = list(fontsize = list(text = 6),
                          strip.background = list(col = "grey90")),
      overlay = list(values = rates_direct,
                     col = "black",
                     lwd = 1.5))
```




```
#Estimates of coverage ratios for census data in eight selected provinces
coverage <- fetch(filename_base,
                  where = c("dataModels", "births_census", "likelihood", "rate")) %>%
    slab(dimension = "province", elements = sample_provinces)

p <- dplot(~  province | age,
           data = coverage,
           col = "grey50",
           horiz = TRUE,
           xlab = "",
           ylab = "",
           layout = c(NA, 1),
           as.table = TRUE,
           scales = list(tck = 0.3),
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))
p <- p + layer(panel.abline(v = 1,
                            col = "black",
                            lty = "dotted"),
               under = FALSE)
plot(p)
```



Fit the revised model

```
filename_revised <- "cambodia_revised.est"
if(!file.exists(filename_revised)) {
    #Get datasets
    births_dhs_data <- CountsOne(bdefdata::cambodia_births_dhs_revised$births,
                                 labels = bdefdata::cambodia_births_dhs_revised$age,
                                 name = "age") %>%
        toInteger(force = TRUE)

    datasets <- list(births_census = births_census,
                     births_dhs = births_dhs_data)

    #Set up system models
    model <- Model(y ~ Poisson(mean ~ age + province),
                   age ~ DLM(damp = NULL),
                   province ~ Exch(covariates = Covariates(mean ~ poverty, data = poverty)),
                   jump = 0.05)

    #Set up data models
    #Data model for census
    model_census <- Model(births_census ~ Poisson(mean ~ age + province),
                          age ~ DLM(trend = NULL,
                                    damp = NULL),
                          province ~ Exch(error = Error(scale = HalfT(scale = 0.05))),
                          jump = 0.07)

    #Data model for DHS
    mean_dhs <- ValuesOne(rep(1, times = nrow(bdefdata::cambodia_births_dhs_revised)),
                          labels = bdefdata::cambodia_births_dhs_revised$age,
                          name = "age")
    se_dhs <- ValuesOne(bdefdata::cambodia_births_dhs_revised$se,
                        labels = bdefdata::cambodia_births_dhs_revised$age,
                        name = "age")

    model_dhs <- Model(births_dhs ~ NormalFixed(mean = mean_dhs, sd = se_dhs))

    data_models <- list(model_dhs, model_census)

    #Estimation
    y <- toInteger(2 * datasets$births_census) # starting values for calculations
```

```
    n_burnin <- 400000
    n_sim <- 300000
    n_chain <-4
    n_thin <- 600

    set.seed(0)

    estimateCounts(model = model,
                   y = y,
                   exposure = exposure_province,
                   dataModels = data_models,
                   datasets = datasets,
                   filename = filename_revised,
                   nBurnin = n_burnin,
                   nSim = n_sim,
                   nChain = n_chain,
                   nThin = n_thin)

}
fetchSummary(filename_revised)
```

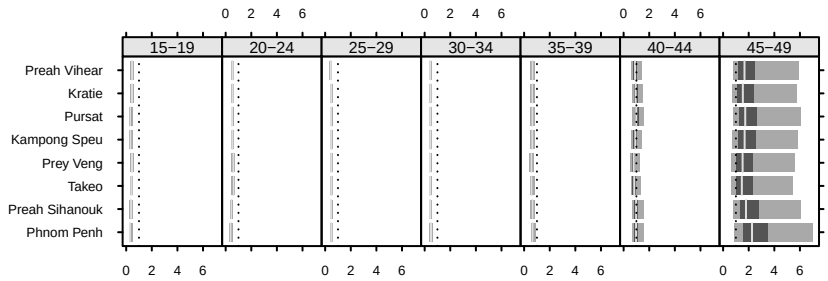
```
## ----------------------------------------------------
## model:
## y ~ Poisson(mean ~ age + province),
## age ~ DLM(damp = NULL),
## province ~ Exch(covariates = Covariates(mean ~ poverty, data = poverty)),
## 0.05
## dimensions: age, province
## ----------------------------------------------------
## y:
## dimensions: age, province
## n cells: 168
## ----------------------------------------------------
## Data models:
## *births_census*
## births_census ~ Poisson(mean ~ age + province),
## age ~ DLM(trend = NULL, damp = NULL),
## province ~ Exch(error = Error(scale = HalfT(scale = 0.05))),
## 0.07
## dimensions: age, province
## *births_dhs*
## births_dhs ~ NormalFixed(mean = mean_dhs, sd = se_dhs)
## dimensions: age
## ----------------------------------------------------
## Datasets:
## *births_census*
## Object of class "Counts"
## dimensions: age, province
## n cells: 168,  n missing: 0, integers: TRUE, n zeros: 0, median: 536.5
## *births_dhs*
## Object of class "Counts"
## dimensions: age
## n cells: 6,  n missing: 0, integers: TRUE, n zeros: 0, median: 43973
```

```
## ------------------------------------------------
## MCMC statistics:
## nBurnin: 400000,  nSim: 300000,  nChain: 4,  nThin: 600,  nIteration: 2000
##
## Metropolis-Hastings updates:
##                                          jump acceptance autocorr
## model.likelihood.rate                    0.05      0.365    0.578
## dataModels.births_census.likelihood.rate 0.07      0.403    0.583
##
## parameters:
##                                                    Rhat    2.5%    50%
## model.likelihood.rate                             . 1.16 0.00312 0.0788
## model.prior.mean                                  . 1.24   -5.79  -2.55
## model.prior.sd                                      1.04 0.00421 0.0767
## model.hyper.age.scaleLevel                          1.00  0.0412  0.405
## model.hyper.age.scaleTrend                          1.01   0.095  0.962
## model.hyper.age.scaleError                          1.00  0.0348  0.311
## model.hyper.province.coef                           1.01   0.359  0.514
## model.hyper.province.scaleError                     1.01  0.0439  0.158
## y                                                 . 1.15      17   1046
## dataModels.births_census.likelihood.rate          . 1.27   0.342  0.551
## dataModels.births_census.prior.mean               . 1.22  -0.983 -0.624
## dataModels.births_census.prior.sd                   1.05  0.0533  0.139
## dataModels.births_census.hyper.age.scaleLevel       1.07   0.193  0.417
## dataModels.births_census.hyper.age.scaleError       1.04   0.024  0.221
## dataModels.births_census.hyper.province.scaleError  1.01 0.00291 0.0467
##                                                   97.5% length
## model.likelihood.rate                             0.279    168
## model.prior.mean                                 -1.26    168
## model.prior.sd                                    0.17      1
## model.hyper.age.scaleLevel                        1.37      1
## model.hyper.age.scaleTrend                         1.8      1
## model.hyper.age.scaleError                        0.856     1
## model.hyper.province.coef                         0.627     1
## model.hyper.province.scaleError                   0.204     1
## y                                                  9548    168
## dataModels.births_census.likelihood.rate           2.39    168
## dataModels.births_census.prior.mean               0.851    168
## dataModels.births_census.prior.sd                 0.179     1
## dataModels.births_census.hyper.age.scaleLevel     0.992     1
## dataModels.births_census.hyper.age.scaleError     0.462     1
## dataModels.births_census.hyper.province.scaleError 0.141    1
## ------------------------------------------------
```

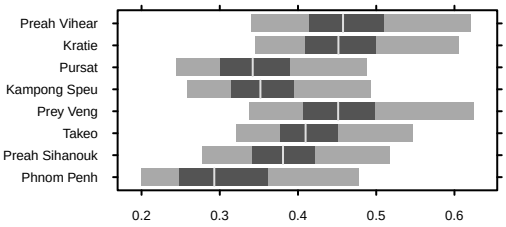
Results of the revised model

```
#Estimates of census coverage ratios from the revised model.
coverage <- fetch(filename_revised,
                  where = c("dataModels", "births_census", "likelihood", "rate")) %>%
    slab(dimension = "province", elements = sample_provinces)

p <- dplot(~  province | age,
           data = coverage,
```

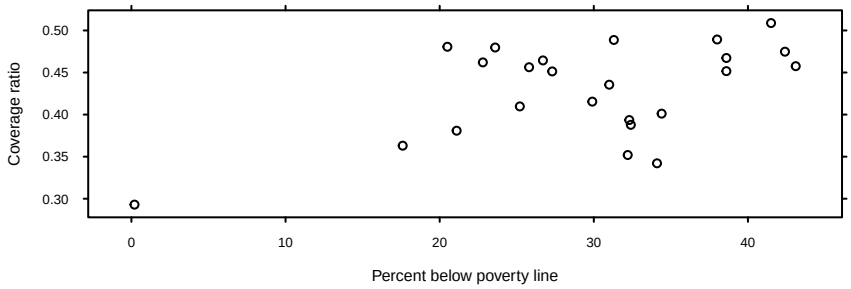
```
            col = "grey50",
            horiz = TRUE,
            xlab = "",
            ylab = "",
            layout = c(NA, 1),
            as.table = TRUE,
            scales = list(tck = 0.3),
            par.settings = list(fontsize = list(text = 6),
                                strip.background = list(col = "grey90")))
p <- p + layer(panel.abline(v = 1,
                            col = "black",
                            lty = "dotted"),
               under = FALSE)
plot(p)
```




```
#Estimates of census coverage ratios for 15-19 year olds from revised model.
coverage <- fetch(filename_revised,
                  where = c("dataModels", "births_census", "likelihood", "rate")) %>%
    subarray(age == "15-19") %>%
    slab(dimension = "province", elements = sample_provinces)

p <- dplot(~  province,
            data = coverage,
            col = "grey50",
            horiz = TRUE,
            xlab = "",
            ylab = "",
            as.table = TRUE,
            scales = list(tck = 0.3),
            par.settings = list(fontsize = list(text = 6),
                                strip.background = list(col = "grey90")))
p <- p + layer(panel.abline(v = 1,
                            col = "black",
                            lty = "dotted"),
               under = FALSE)
plot(p)
```




```
#Estimates of coverage ratios for census data in eight selected provinces.
cover <- fetch(filename_revised,
               where = c("dataModels", "births_census", "likelihood", "rate")) %>%
    subarray(age == "15-19") %>%
    collapseIterations(FUN = median) %>%
    as.data.frame(direction = "long", stringsAsFactors = FALSE) %>%
    left_join(poverty, by = "province")

p <- xyplot(value ~ poverty,
            data = cover,
            scales = list(tck = 0.3),
            par.settings = list(fontsize = list(text = 6, points = 5)),
            col = "black",
            xlab = "Percent below poverty line",
            ylab = "Coverage ratio")
plot(p)
```



Fit the final model

```
filename_final <- "cambodia_final.est"
if(!file.exists(filename_final)) {
    #Get datasets
    births_dhs_data <- CountsOne(bdefdata::cambodia_births_dhs_revised$births,
                                 labels = bdefdata::cambodia_births_dhs_revised$age,
                                 name = "age") %>%
        toInteger(force = TRUE)

    datasets <- list(births_census = births_census,
                     births_dhs = births_dhs_data)

    #Set up system models
```

```
    pp1519 <- datasets$births_census %>%
      dimnames() %>%
      expand.grid() %>%
      mutate(is_pp1519 = 1 * (age == "15-19" & province == "Phnom Penh"))

    model <- Model(y ~ Poisson(mean ~ age * province),
                   age ~ DLM(damp = NULL),
                   province ~ Exch(covariates = Covariates(mean ~ poverty, data = poverty)),
                   age:province ~ Exch(covariates = Covariates(mean ~ is_pp1519, data = pp1519)),
                   jump = 0.05)

    #Set up data models
    #Data model for census
    model_census <- Model(births_census ~ Poisson(mean ~ age + province),
                          age ~ DLM(trend = NULL,
                                    damp = NULL),
                          province ~ Exch(error = Error(scale = HalfT(scale = 0.05))),
                          jump = 0.07)

    #Data model for DHS
    mean_dhs <- ValuesOne(rep(1, times = nrow(bdefdata::cambodia_births_dhs_revised)),
                          labels = bdefdata::cambodia_births_dhs_revised$age,
                          name = "age")
    se_dhs <- ValuesOne(bdefdata::cambodia_births_dhs_revised$se,
                        labels = bdefdata::cambodia_births_dhs_revised$age,
                        name = "age")

    model_dhs <- Model(births_dhs ~ NormalFixed(mean = mean_dhs, sd = se_dhs))

    data_models <- list(model_dhs, model_census)

    #Estimation
    y <- toInteger(2 * datasets$births_census) # starting values for calculations

    n_burnin <- 400000
    n_sim <- 400000
    n_chain <-4
    n_thin <- 800

    set.seed(0)

    estimateCounts(model = model,
                   y = y,
                   exposure = exposure_province,
                   dataModels = data_models,
                   datasets = datasets,
                   filename = filename_final,
                   nBurnin = n_burnin,
                   nSim = n_sim,
                   nChain = n_chain,
                   nThin = n_thin)

}
```

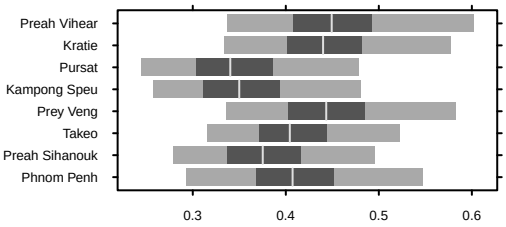
```
fetchSummary(filename_final)
```

```
## ----------------------------------------------------
## model:
## y ~ Poisson(mean ~ age * province),
## age ~ DLM(damp = NULL),
## province ~ Exch(covariates = Covariates(mean ~ poverty, data = poverty)),
## age:province ~ Exch(covariates = Covariates(mean ~ is_pp1519,
##     data = pp1519)),
## 0.05
## dimensions: age, province
## ----------------------------------------------------
## y:
## dimensions: age, province
## n cells: 168
## ----------------------------------------------------
## Data models:
## *births_census*
## births_census ~ Poisson(mean ~ age + province),
## age ~ DLM(trend = NULL, damp = NULL),
## province ~ Exch(error = Error(scale = HalfT(scale = 0.05))),
## 0.07
## dimensions: age, province
## *births_dhs*
## births_dhs ~ NormalFixed(mean = mean_dhs, sd = se_dhs)
## dimensions: age
## ----------------------------------------------------
## Datasets:
## *births_census*
## Object of class "Counts"
## dimensions: age, province
## n cells: 168,  n missing: 0, integers: TRUE, n zeros: 0, median: 536.5
## *births_dhs*
## Object of class "Counts"
## dimensions: age
## n cells: 6,  n missing: 0, integers: TRUE, n zeros: 0, median: 43973
## ----------------------------------------------------
## MCMC statistics:
## nBurnin: 400000,  nSim: 400000,  nChain: 4,  nThin: 800,  nIteration: 2000
##
## Metropolis-Hastings updates:
##                                          jump acceptance autocorr
## model.likelihood.rate                    0.05      0.374    0.560
## dataModels.births_census.likelihood.rate 0.07      0.392    0.514
##
## parameters:
##                                                Rhat    2.5%    50%
## model.likelihood.rate                        . 1.14 0.00307 0.0799
## model.prior.mean                             . 1.11   -5.78  -2.52
## model.hyper.age.scaleLevel                     1.00  0.0242  0.638
## model.hyper.age.scaleTrend                     1.01   0.134  0.548
## model.hyper.age.scaleError                     1.00  0.0726  0.282
## model.hyper.province.coef                      1.00   0.425  0.498
## model.hyper.province.scaleError                1.02   0.115  0.149
```

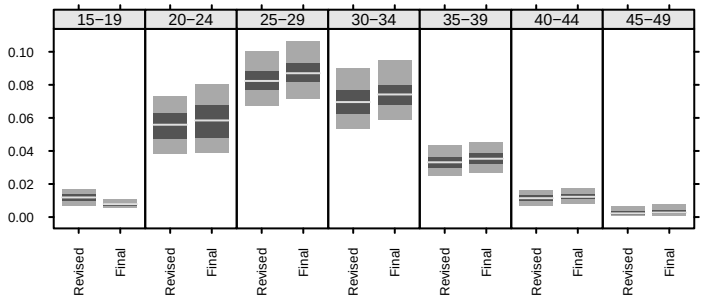
```
## model.hyper.age:province.coef                          1.00   -1.05 -0.743
## model.hyper.age:province.scaleError                    1.03  0.0115  0.068
## y                                                    . 1.13      19   1032
## dataModels.births_census.likelihood.rate             . 1.10    0.34  0.543
## dataModels.births_census.prior.mean                  . 1.11   -0.98 -0.629
## dataModels.births_census.prior.sd                      1.04  0.0397  0.122
## dataModels.births_census.hyper.age.scaleLevel          1.02  0.0463  0.334
## dataModels.births_census.hyper.age.scaleError          1.00 0.00607  0.179
## dataModels.births_census.hyper.province.scaleError     1.03 0.00754 0.0375
##                                                      97.5% length
## model.likelihood.rate                                0.279    168
## model.prior.mean                                     -1.27    168
## model.hyper.age.scaleLevel                            1.91      1
## model.hyper.age.scaleTrend                             1.5      1
## model.hyper.age.scaleError                             1.4      1
## model.hyper.province.coef                             0.63      1
## model.hyper.province.scaleError                      0.231      1
## model.hyper.age:province.coef                       -0.366      1
## model.hyper.age:province.scaleError                  0.153      1
## y                                                     9558    168
## dataModels.births_census.likelihood.rate              2.71    168
## dataModels.births_census.prior.mean                   1.01    168
## dataModels.births_census.prior.sd                    0.178      1
## dataModels.births_census.hyper.age.scaleLevel        0.886      1
## dataModels.births_census.hyper.age.scaleError        0.835      1
## dataModels.births_census.hyper.province.scaleError 0.0925      1
## ------------------------------------------------------
```

Results of the final model

```
#Estimates of census coverage ratios for 15-19 year olds from final model.
coverage <- fetch(filename_final,
                  where = c("dataModels", "births_census", "likelihood", "rate")) %>%
    subarray(age == "15-19") %>%
    slab(dimension = "province", elements = sample_provinces)

p <- dplot(~  province,
           data = coverage,
           col = "grey50",
           horiz = TRUE,
           xlab = "",
           ylab = "",
           as.table = TRUE,
           scales = list(tck = 0.3),
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))
p <- p + layer(panel.abline(v = 1,
                            col = "black",
                            lty = "dotted"),
               under = FALSE)
plot(p)
```

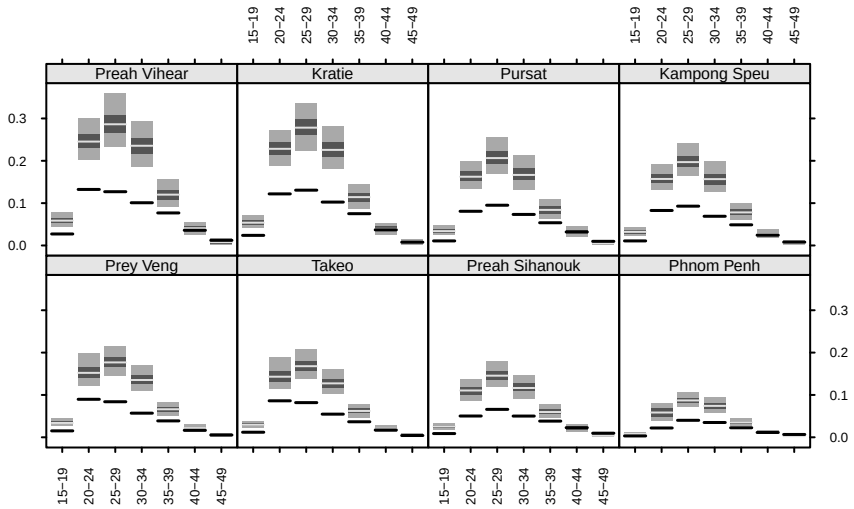



```
#Estimates of age-specific fertility rates in Phnom Penh from revised and final models.
rates <- dbind(Revised = fetch(filename_revised, where = c("model", "likelihood", "rate")),
               Final = fetch(filename_final, where = c("model", "likelihood", "rate")),
               along = "model") %>%
    subarray(province == "Phnom Penh")

p <- dplot(~ model | age,
           data = rates,
           col = "grey50",
           xlab = "",
           ylab = "",
           as.table = TRUE,
           scales = list(tck = 0.3,
                         x = list(rot = 90, alternating = FALSE)),
           layout = c(NA, 1),
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))
plot(p)
```

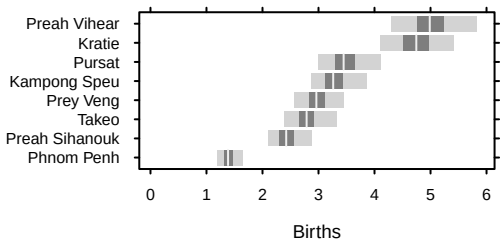



```
#Estimates of age-specific fertility rates for eight selected provinces
rates <- fetch(filename_final,
               where = c("model", "likelihood", "rate")) %>%
    slab(dimension = "province", elements = rev(sample_provinces))

births_census <- fetch(filename_base,
                       where = c("datasets", "births_census"))
exposure <- fetch(filename_base,
                  where = "exposure")
rates_direct <- (births_census / exposure_province) %>%
    subarray(province %in% sample_provinces) %>%
    slab(dimension = "province", elements = rev(sample_provinces))
```

```
dplot(~ age | province,
      data = rates,
      col = "grey50",
      xlab = "",
      ylab = "",
      as.table = TRUE,
      scales = list(tck = 0.3,
                    x = list(rot = 90)),
      layout = c(NA, 2),
      par.settings = list(fontsize = list(text = 6),
                          strip.background = list(col = "grey90")),
      overlay = list(values = rates_direct,
                     col = "black",
                     lwd = 1.5))
```




```
#Estimates of age-specific fertility rates for eight selected provinces
tfr <- fetch(filename_final, where = c("model", "likelihood", "rate")) %>%
    tfr()

tfr <- tfr %>%
    slab(dimension = "province", elements = sample_provinces)

dplot(~ province,
      data = tfr,
      col = "grey",
      xlab = "Births",
      ylab = "",
      horizontal = TRUE,
      xlim = c(-0.2, NA),
      scales = list(tck = 0.3),
      layout = c(NA, 1),
      par.settings = list(fontsize = list(text = 7),
                          strip.background = list(col = "grey90")))
```


Preah Vihear
Kratie
Pursat
Kampong Speu
Prey Veng
Takeo
Preah Sihanouk
Phnom Penh
0
1
2
3
4
5
6
Births