Chapter 19 Population in China

John Bryant and Junni L. Zhang

2018-09-10

Setup

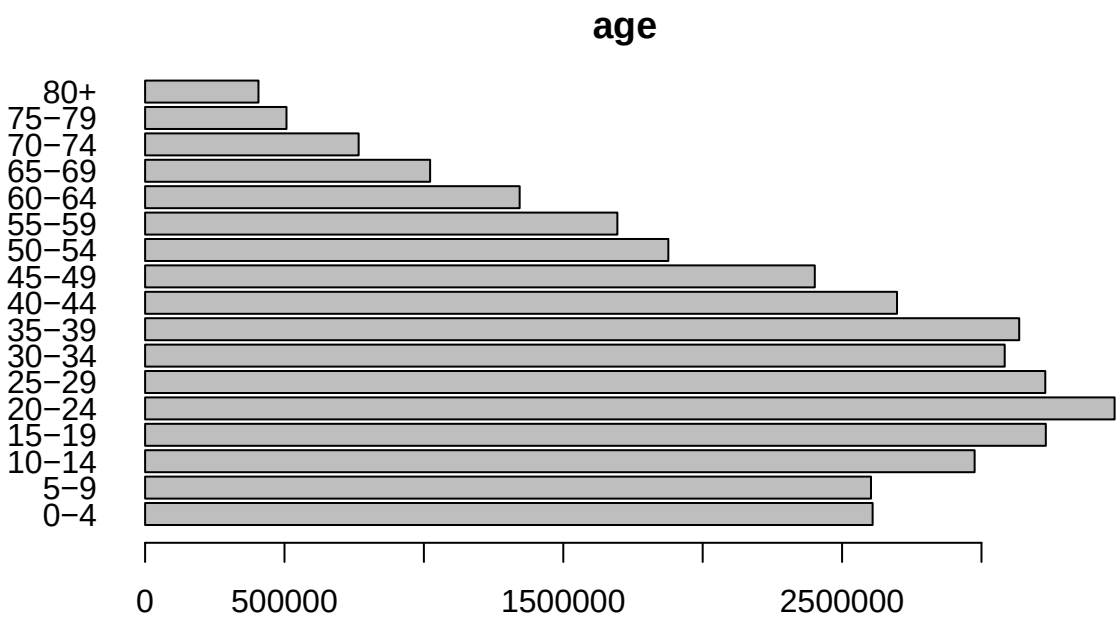
```
library(methods)
library(bdefdata)
library(demest)
library(demlife)
library(dplyr)
library(tidyverse)
library(latticeExtra)
```

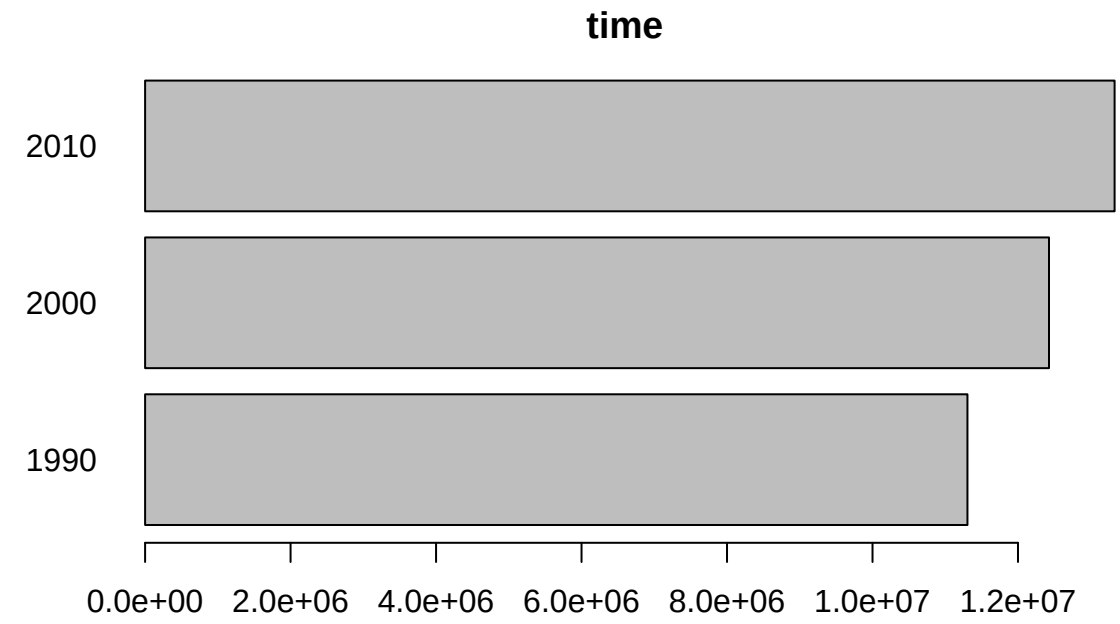
Get data

```
census <- bdefdata::china_population_census %>%
  Counts()
summary(census)
```

```
##
## name:     age       time
## length:   17        3
## dimtype:  age       time
## dimscale: Intervals Points
## first:    0-4       1990
## last:     80+       2010
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   76764  414086  787532  726654 1020226 1274125
```

```
plot(census)
```



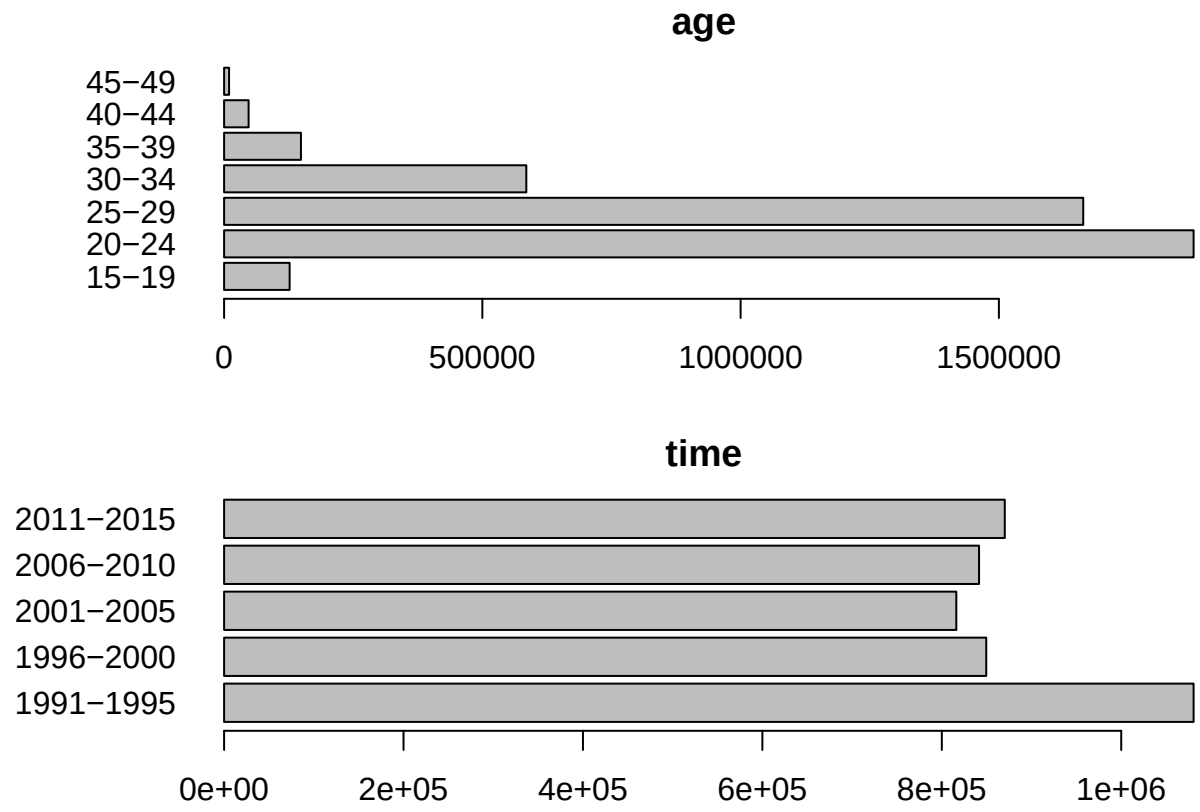



```
births_un <- bdefdata::china_births_un %>%
  Counts()
summary(births_un)
```

```
##
## name:     age       time
## length:   7         5
## dimtype:  age       time
## dimscale: Intervals Intervals
## first:    15-19     1991-1995
## last:     45-49     2011-2015
```

```
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1330   13835   39630  127373  292470  487940
```

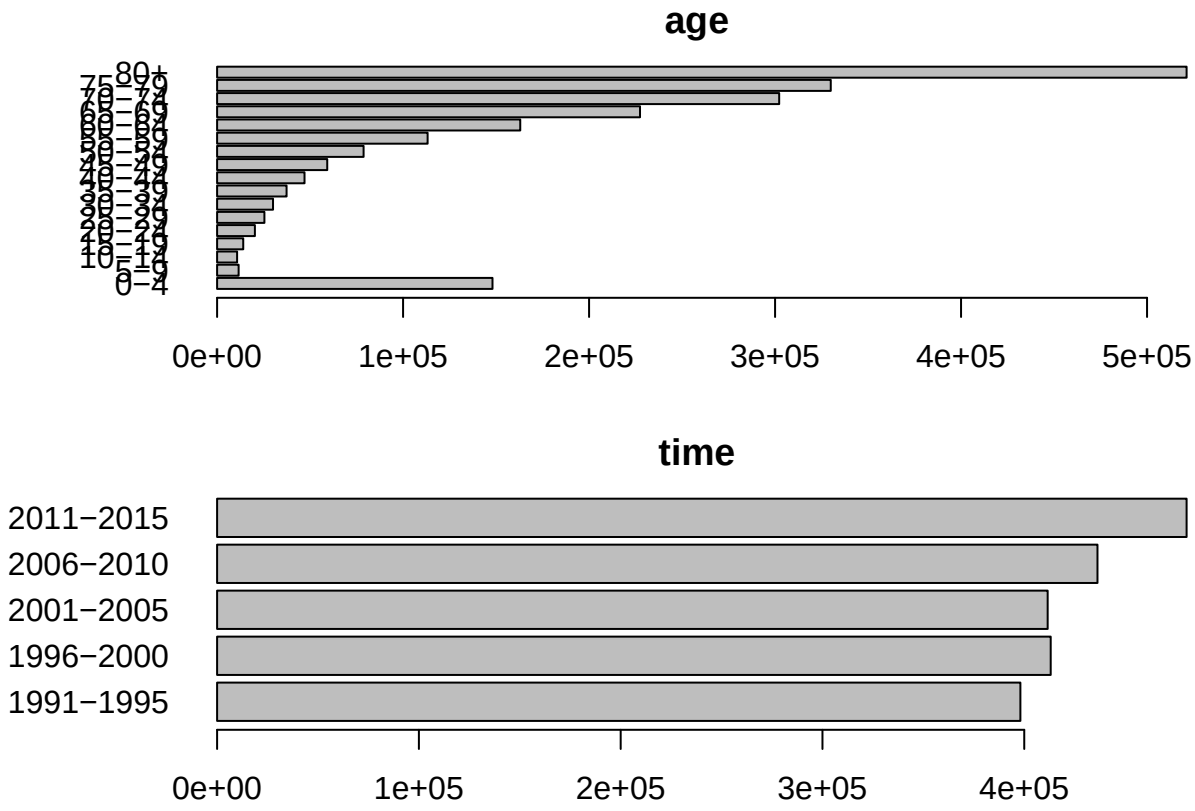
```
plot(births_un)
```




```
deaths_un <- bdefdata::china_deaths_un %>%
  Counts()
summary(deaths_un)
```

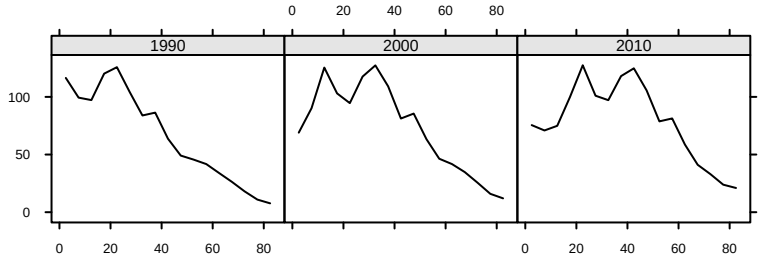
```
##
## name:     age       time
## length:   17        5
## dimtype:  age       time
## dimscale: Intervals Intervals
## first:    0-4       1991-1995
## last:     80+       2011-2015
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    4740   11990   25171   40510  141700
```

```
plot(deaths_un)
```

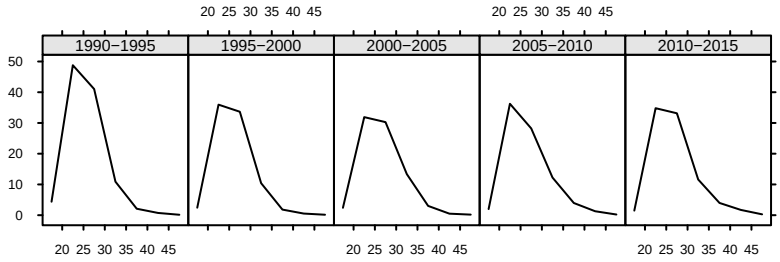


Plot data

```
#Plot census counts of population (millions), by age.
denom_china <- 100
census_df <- census %>%
    as.data.frame(direction = "long", midpoints = "age") %>%
    mutate(count = 1e-6 * denom_china * count)

xyplot(count ~ age | factor(time),
       data = census_df,
       col = "black",
       type = "l",
       layout = c(NA, 1),
       xlab = "",
       ylab = "",
       prepanel = function(y) list(ylim = c(0, max(y, na.rm = TRUE))),
       scales = list(tck = 0.4),
       as.table = TRUE,
       par.settings = list(fontsize = list(text = 6),
                           strip.background = list(col = "grey90")))
```

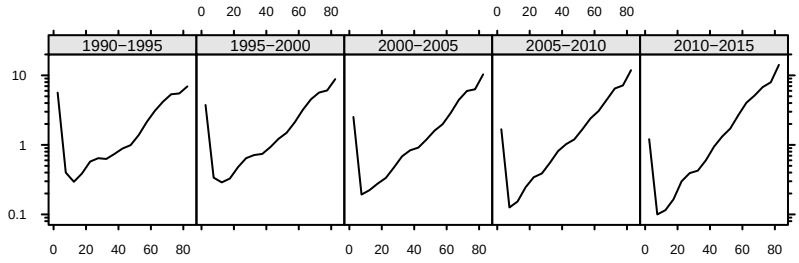



```
#Plot estimates of births (millions), by age of mother.
births_df <- births_un %>%
    as.data.frame(direction = "long", midpoints = "age") %>%
    mutate(count = 1e-6 * denom_china * count) %>%
    mutate(time = gsub("1\\-", "0-", time),
           time = gsub("6\\-", "5-", time))
xyplot(count ~ age | time,
       data = births_df,
       col = "black",
       xlab = "",
       ylab = "",
       type = "l",
       layout = c(NA, 1),
       scales = list(tck = 0.3),
       as.table = TRUE,
       par.settings = list(fontsize = list(text = 6),
                           strip.background = list(col = "grey90")))
```



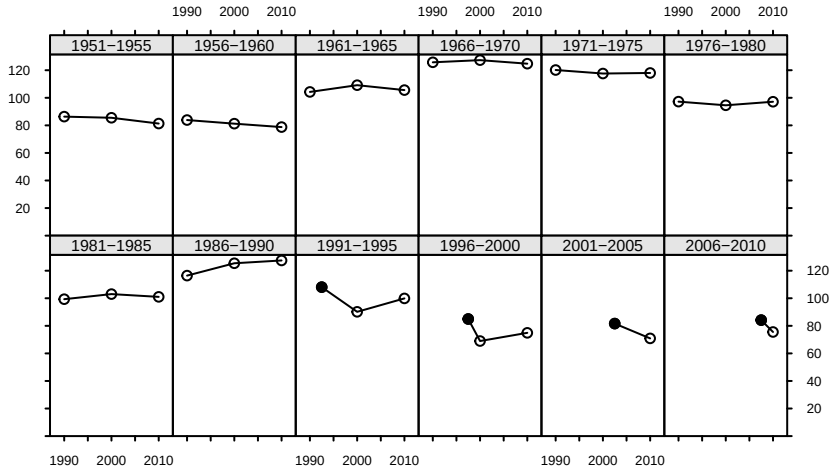

```
#Plot estimates of deaths (millions), by age. The data are shown on a log scale.
deaths_df <- deaths_un  %>%
    as.data.frame(direction = "long", midpoints = "age") %>%
    mutate(count = 1e-6 * denom_china * count) %>%
    mutate(time = gsub("1\\-", "0-", time),
           time = gsub("6\\-", "5-", time))

lty <- c("solid", "dotted")
xyplot(count ~ age | time,
       data = deaths_df,
       col = "black",
       type = "l",
       lty = lty,
       xlab = "",
```

```
       ylab = "",
       lwd = 1,
       midpoints = "age",
       layout = c(NA, 1),
       scales = list(x = list(tck = 0.3),
                     y = list(log = TRUE, tck = 0.45)),
       yscale.components = yscale.components.log10ticks,
       as.table = TRUE,
       par.settings = list(fontsize = list(text = 6),
                           strip.background = list(col = "grey90")))
```




```
#Plot changes in cohort size (millions), as implied by the births and census data.
census_df <- census  %>%
          as.data.frame(direction = "long", midpoints = "age") %>%
          mutate(cohort = time - age) %>%
          mutate(count = 1e-6 * 100 * count)

births_df <- births_un %>%
             collapseDimension(dimension="age") %>%
             as.data.frame(direction = "long") %>%
             mutate(count = 1e-6 * 100 * count) %>%
             mutate(time = c(1992.5, 1997.5, 2002.5, 2007.5, 2012.5)) %>%
             mutate(cohort = time)

cohorts <- rbind(census_df[census_df$cohort>=1932.5,2:4],births_df[births_df$cohort<=2007.5,]) %>%
           mutate(cohort.name = sprintf("%s-%s",round(cohort-1.5),round(cohort+2.5))) %>%
           arrange(cohort.name,time)

cohorts.births <- cohorts
cohorts.births$count[cohorts.births$time != cohorts.births$cohort] <- NA

p <- xyplot(count ~ time | factor(cohort.name),
       data = cohorts,
       subset = cohort>=1952.5,
       type = "b",
       col = "black",
       xlab = "",
       ylab = "",
       xlim = c(1987, 2013),
       ylim = c(0,NA),
       as.table = TRUE,
       layout = c(NA,2),
       scales = list(tck = 0.3),
```

```
       par.settings = list(strip.background = list(col = "grey90"),
                           fontsize = list(points=6, text=6)))
p <- p + as.layer(xyplot(count ~ time | factor(cohort.name),
       data = cohorts.births,
       subset = cohort>=1952.5,
       type = "p",
       pch = 19,
       col = "black"))

plot(p)
```



Fit model

```
filename <- "china_model.est"
if(!file.exists(filename)) {

    #Initial account
    population <- dbind(extrapolate(subarray(census, time == "1990", drop = FALSE),
                                    labels = "1995"),
                        extrapolate(subarray(census, time == "2000", drop = FALSE),
                                    labels = "2005"),
                        extrapolate(subarray(census, time == "2010", drop = FALSE),
                                    labels = c("2015", "2020")),
                        along = "time")

    births <- extrapolate(births_un, labels = "2016-2020")

    deaths <- extrapolate(deaths_un, labels = "2016-2020")

    account <- Movements(population = population,
                         births = births,
                         exits = list(deaths = deaths)) %>%
        makeConsistent()

    #Get datasets.
```

```
datasets <- list(census = census,
                 births_un = births_un,
                 deaths_un = deaths_un)

#Set up data models.
mean <- census / census
sd <- 0.005 * as(census, "Values")

census_dmodel <- Model(census ~ NormalFixed(mean = mean, sd = sd),
                       series = "population")

births_un_dmodel <- Model(births_un ~ Poisson(mean ~ 1),
                       `(Intercept)` ~ ExchFixed(sd = 0.1),
                       series = "births",
                       priorSD = HalfT(scale = 0.1),
                       jump = 0.01)

deaths_un_dmodel <- Model(deaths_un ~ Poisson(mean ~ 1),
                       `(Intercept)` ~ ExchFixed(sd = 0.1),
                       series = "deaths",
                       priorSD = HalfT(scale = 0.1),
                       jump = 0.03)

dataModels <- list(census_dmodel,
                   births_un_dmodel,
                   deaths_un_dmodel)

#Set up system models
population <- Model(population ~ Poisson(mean ~ age * time,
                                          useExpose = FALSE),
                    age ~ DLM(damp = NULL),
                    time ~ DLM(damp = NULL),
                    age:time ~ DLM(damp = NULL),
                    jump = 0.005)

births <- Model(births ~ Poisson(mean ~ age * time),
                age ~ DLM(damp = NULL),
                time ~ DLM(level = Level(scale = HalfT(scale = 0.1)),
                           trend = NULL,
                           damp = NULL),
                age:time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                               trend = NULL,
                               damp = NULL),
                upper = 0.1,
                jump = 0.015)

deaths <- Model(deaths ~ Poisson(mean ~ age * time),
                age ~ DLM(damp = NULL),
                time ~ DLM(level = Level(scale = HalfT(scale = 0.1)),
                           trend = Trend(scale = HalfT(scale = 0.1)),
                           damp = NULL),
                age:time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                               trend = NULL,
```

```
                                    damp = NULL),
                    upper = 1,
                    jump = 0.03)

    systemModels <- list(population,
                         births,
                         deaths)

    #Estimate account.
    n_burnin <- 700000
    n_sim <- 800000
    n_chain <- 4
    n_thin <- 400

    set.seed(0)

    estimateAccount(account = account,
                    datasets = datasets,
                    dataModels = dataModels,
                    systemModels = systemModels,
                    filename = filename,
                    nBurnin = n_burnin,
                    nSim = n_sim,
                    nChain = n_chain,
                    nThin = n_thin)

    options(width = 150)
}
fetchSummary(filename)
```

```
## ------------------------------------------------
## Account:
## *population*
## dimensions: age, time
## n cells: 119
## *births*
## dimensions: age, time, triangle
## n cells: 84
## *deaths*
## dimensions: age, time, triangle
## n cells: 204
## ------------------------------------------------
## System models:
## *population*
## population ~ Poisson(mean ~ age * time, useExpose = FALSE),
## age ~ DLM(damp = NULL),
## time ~ DLM(damp = NULL),
## age:time ~ DLM(damp = NULL),
## 0.005
## dimensions: age, time
## *births*
## births ~ Poisson(mean ~ age * time),
## age ~ DLM(damp = NULL),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.1)), trend = NULL,
```

```
##     damp = NULL),
## age:time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = NULL,
##     damp = NULL),
## 0.1,
## 0.015
## dimensions: age, time, triangle
## *deaths*
## deaths ~ Poisson(mean ~ age * time),
## age ~ DLM(damp = NULL),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.1)), trend = Trend(scale = HalfT(scale = 0.1)),
##     damp = NULL),
## age:time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = NULL,
##     damp = NULL),
## 1,
## 0.03
## dimensions: age, time, triangle
## ------------------------------------------------
## Datasets:
## *census*
## Object of class "Counts"
## dimensions: age, time
## n cells: 51,  n missing: 0, integers: TRUE, n zeros: 0, median: 787532
## *births_un*
## Object of class "Counts"
## dimensions: age, time
## n cells: 35,  n missing: 0, integers: TRUE, n zeros: 0, median: 39630
## *deaths_un*
## Object of class "Counts"
## dimensions: age, time
## n cells: 85,  n missing: 0, integers: TRUE, n zeros: 0, median: 11990
## ------------------------------------------------
## Data models:
## *census*
## census ~ NormalFixed(mean = mean, sd = sd),
## "population"
## dimensions: age, time
## *births_un*
## births_un ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.1),
## HalfT(scale = 0.1),
## 0.01,
## "births"
## dimensions: age, time
## *deaths_un*
## deaths_un ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.1),
## HalfT(scale = 0.1),
## 0.03,
## "deaths"
## dimensions: age, time
## ------------------------------------------------
## MCMC statistics:
## nBurnin: 700000,  nSim: 800000,  nChain: 4,  nThin: 400,  nIteration: 8000
##
```

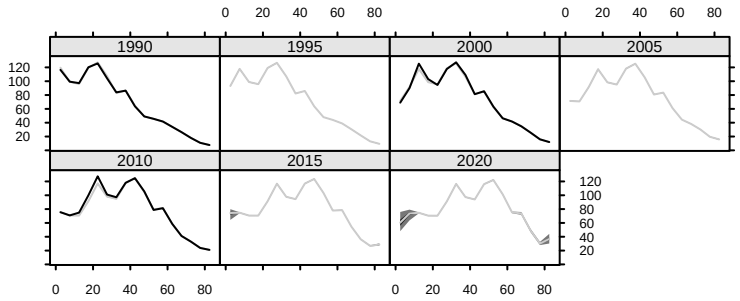
```
## Metropolis-Hastings updates:
##                                          jump acceptance autocorr
## systemModels.population.likelihood.count 0.005      0.274    0.347
## systemModels.births.likelihood.rate      0.015      0.308    0.864
## systemModels.deaths.likelihood.rate      0.030      0.308    0.582
## dataModels.births_un.likelihood.rate     0.010      0.320    0.681
## dataModels.deaths_un.likelihood.rate     0.030      0.253    0.623
##
## parameters:
##                                                 Rhat     2.5%     50%
## account.population                                . 1.10   113260  778270
## account.births                                    . 1.24      610   14929
## account.deaths                                    . 1.18      484    4920
## systemModels.population.likelihood.count          . 1.16   113685  778068
## systemModels.population.prior.mean                  1.08     11.7    13.6
## systemModels.population.hyper.age.scaleLevel        1.00   0.0551   0.451
## systemModels.population.hyper.age.scaleTrend        1.01   0.0175   0.246
## systemModels.population.hyper.age.scaleError        1.00   0.0428    0.49
## systemModels.population.hyper.time.scaleLevel       1.00  0.00159  0.0209
## systemModels.population.hyper.time.scaleTrend       1.00  0.00619  0.0226
## systemModels.population.hyper.time.scaleError       1.00 0.000816  0.0134
## systemModels.population.hyper.age:time.scaleLevel   1.00   0.0461   0.107
## systemModels.population.hyper.age:time.scaleTrend   1.00  0.00695  0.0264
## systemModels.population.hyper.age:time.scaleError   1.01   0.0279  0.0835
## systemModels.births.likelihood.rate               . 1.14 0.000334 0.00547
## systemModels.births.prior.mean                    . 1.14    -7.98   -5.25
## systemModels.births.prior.sd                      . 1.17   0.0443    0.16
## systemModels.births.hyper.age.scaleLevel            1.01    0.103   0.731
## systemModels.births.hyper.age.scaleTrend            1.01   0.0594   0.503
## systemModels.births.hyper.age.scaleError            1.02  0.00805   0.576
## systemModels.births.hyper.time.scaleLevel           1.00  0.00642  0.0696
## systemModels.births.hyper.time.scaleError           1.01   0.0568   0.163
## systemModels.births.hyper.age:time.scaleLevel       1.04  0.00603   0.109
## systemModels.births.hyper.age:time.scaleError       1.00   0.0029  0.0357
## systemModels.deaths.likelihood.rate               . 1.19 0.000269 0.00231
## systemModels.deaths.prior.mean                    . 1.19    -8.21   -6.08
## systemModels.deaths.prior.sd                        1.06   0.0074  0.0612
## systemModels.deaths.hyper.age.scaleLevel            1.08    0.065   0.937
## systemModels.deaths.hyper.age.scaleTrend            1.05   0.0489   0.411
## systemModels.deaths.hyper.age.scaleError          . 1.19   0.0982   0.706
## systemModels.deaths.hyper.time.scaleLevel           1.00  0.00532   0.049
## systemModels.deaths.hyper.time.scaleTrend           1.00  0.00854  0.0511
## systemModels.deaths.hyper.time.scaleError           1.02  0.00124  0.0288
## systemModels.deaths.hyper.age:time.scaleLevel       1.00  0.00442  0.0202
## systemModels.deaths.hyper.age:time.scaleError       1.00  0.00383  0.0155
## dataModels.births_un.likelihood.rate                1.09    0.983    1.14
## dataModels.births_un.prior.mean                     1.01    0.104   0.127
## dataModels.births_un.prior.sd                       1.05   0.0245  0.0546
## dataModels.deaths_un.likelihood.rate                1.03    0.954    1.17
## dataModels.deaths_un.prior.mean                     1.01    0.104   0.166
## dataModels.deaths_un.prior.sd                       1.02   0.0985   0.123
##                                                  97.5% length
## account.population                              1256804    119
## account.births                                   201329     84
```

```
## account.deaths                                        54006    204
## systemModels.population.likelihood.count            1257200    119
## systemModels.population.prior.mean                       14    119
## systemModels.population.hyper.age.scaleLevel           1.59      1
## systemModels.population.hyper.age.scaleTrend           1.32      1
## systemModels.population.hyper.age.scaleError           1.32      1
## systemModels.population.hyper.time.scaleLevel         0.186      1
## systemModels.population.hyper.time.scaleTrend         0.131      1
## systemModels.population.hyper.time.scaleError         0.168      1
## systemModels.population.hyper.age:time.scaleLevel     0.169      1
## systemModels.population.hyper.age:time.scaleTrend    0.0623      1
## systemModels.population.hyper.age:time.scaleError     0.505      1
## systemModels.births.likelihood.rate                  0.0725     84
## systemModels.births.prior.mean                       -2.71     84
## systemModels.births.prior.sd                         0.357      1
## systemModels.births.hyper.age.scaleLevel              1.94      1
## systemModels.births.hyper.age.scaleTrend              1.74      1
## systemModels.births.hyper.age.scaleError               1.8      1
## systemModels.births.hyper.time.scaleLevel            0.198      1
## systemModels.births.hyper.time.scaleError             0.35      1
## systemModels.births.hyper.age:time.scaleLevel        0.208      1
## systemModels.births.hyper.age:time.scaleError        0.152      1
## systemModels.deaths.likelihood.rate                  0.137    204
## systemModels.deaths.prior.mean                       -2.02    204
## systemModels.deaths.prior.sd                         0.136      1
## systemModels.deaths.hyper.age.scaleLevel              2.01      1
## systemModels.deaths.hyper.age.scaleTrend              1.23      1
## systemModels.deaths.hyper.age.scaleError              1.84      1
## systemModels.deaths.hyper.time.scaleLevel              0.15      1
## systemModels.deaths.hyper.time.scaleTrend             0.187      1
## systemModels.deaths.hyper.time.scaleError             0.251      1
## systemModels.deaths.hyper.age:time.scaleLevel        0.0619      1
## systemModels.deaths.hyper.age:time.scaleError        0.0582      1
## dataModels.births_un.likelihood.rate                   1.35     35
## dataModels.births_un.prior.mean                       0.183     35
## dataModels.births_un.prior.sd                         0.156      1
## dataModels.deaths_un.likelihood.rate                   1.62     85
## dataModels.deaths_un.prior.mean                       0.225     85
## dataModels.deaths_un.prior.sd                         0.162      1
## -------------------------------------------------------------
```

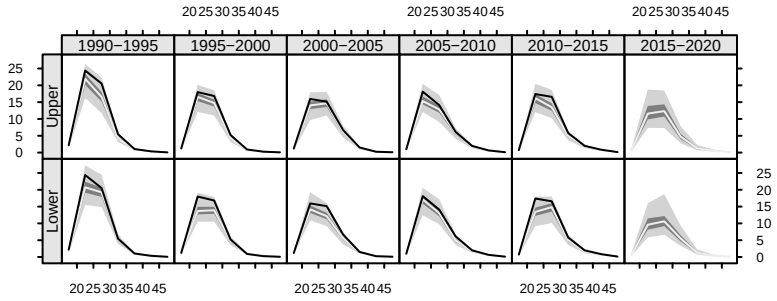
Results

```
#Population counts
population <- fetch(filename,
                    where = c("account", "population"))
population <- 1e-6 * denom_china * population

census_data <- fetch(filename,
                 where = c("datasets", "census"))
census_data <- 1e-6 * denom_china * census_data

p <- dplot(~ age | factor(time),
```

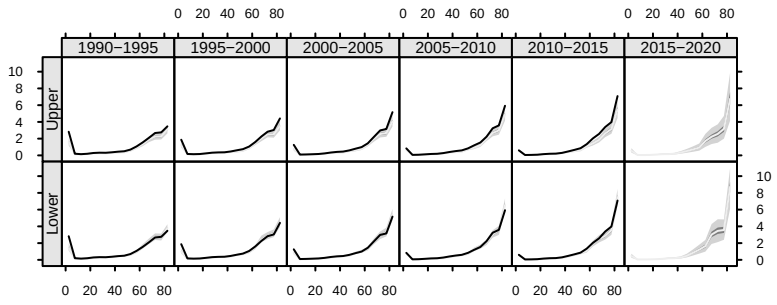
```
            data = population,
            col = "grey20",
            xlab = "",
            ylab = "",
            midpoints = "age",
            as.table = TRUE,
            scales = list(tck = 0.3),
            par.settings = list(fontsize = list(text = 6),
                                strip.background = list(col = "grey90")),
            overlay = list(values = census_data,
                           col = "black"))
plot(p)
```




```
#Birth counts
period_labels <- paste(seq(1990, 2015, 5), seq(1995, 2020, 5), sep = "-")

births <- 1e-6 * denom_china * fetch(filename,
                                     where = c("account", "births")) %>%
    as("array")
dimnames(births)[[3]] <- c("Lower", "Upper")
births <- Counts(births)
births <- births[,,2:1,]
births_data <- 0.5 * 1e-6 * denom_china * fetch(filename,
                                                where = c("datasets", "births"))
p <- dplot(~ age | factor(time) * triangle,
            data = births,
            weights = weights,
            col = "grey",
            xlab = "",
            ylab = "",
            midpoints = "age",
            as.table = TRUE,
            scales = list(tck = 0.3),
            strip = strip.custom(factor.levels = period_labels),
            par.settings = list(fontsize = list(text = 6),
                                strip.background = list(col = "grey90")),
            overlay = list(values = births_data,
                           col = "black",
                           type = "l"))
p <- useOuterStrips(p)
p$condlevels[[1]] <- period_labels
plot(p)
```

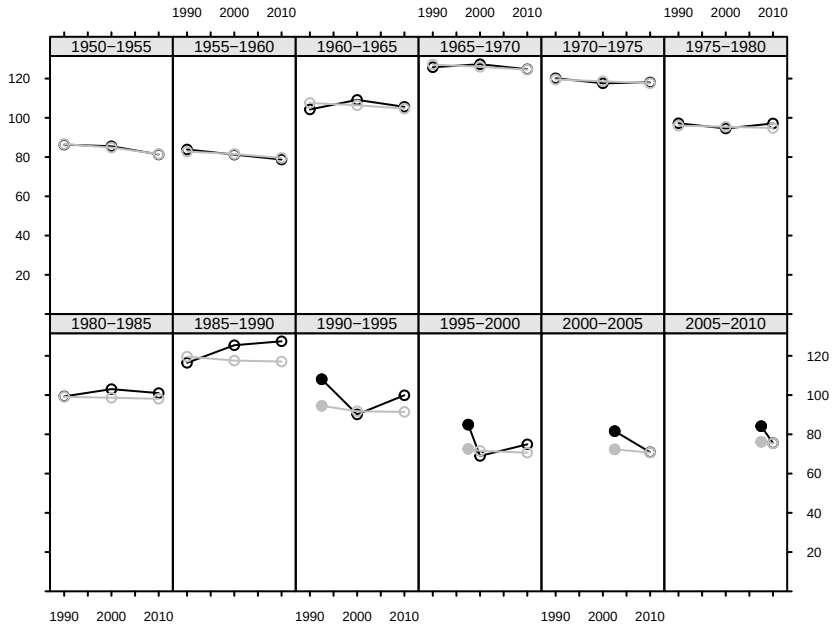



```
#Death counts
period_labels <- paste(seq(1990, 2015, 5), seq(1995, 2020, 5), sep = "-")

deaths <- (1e-6 * denom_china * fetch(filename,
                                      where = c("account", "deaths"))) %>%
    as("array")
dimnames(deaths)[[3]] <- c("Lower", "Upper")
deaths <- deaths[,,2:1,]
deaths <- Counts(deaths)

deaths_data <- (0.5 * 1e-6 * denom_china * fetch(filename,
                                                 where = c("datasets", "deaths_un")))

p <- dplot(~ age | factor(time) * triangle,
           data = deaths,
           col = "grey",
           xlab = "",
           ylab = "",
           midpoints = "age",
           as.table = TRUE,
           scales = list(tck = 0.3),
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")),
           overlay = list(values = deaths_data,
                          col = "black"))
p <- useOuterStrips(p)
p$condlevels[[1]] <- period_labels
plot(p)
```

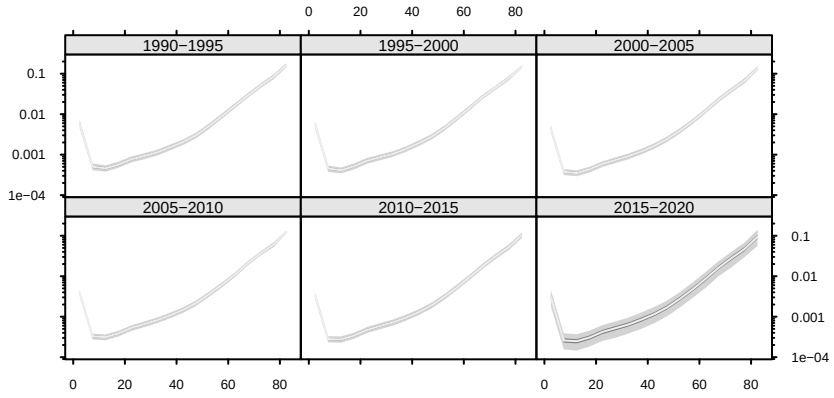


```r
#Change in cohort size
period_labels <- paste(seq(1950, 2005, 5), seq(1955, 2010, 5), sep = "-")

population <- fetch(filename,
                    where = c("account", "population"))
births <- fetch(filename,
                where = c("account", "births")) %>%
    collapseDimension(dimension = c("age", "triangle"))

population.pmean <- collapseIterations(population, FUN = mean) %>%
    as.data.frame(direction = "long", midpoints = "age") %>%
    mutate(cohort = time - age) %>%
    mutate(count = 1e-6 * denom_china * count)

births.pmean <- collapseIterations(births, FUN = mean) %>%
    as.data.frame(direction = "long") %>%
    mutate(time = c(1992.5, 1997.5, 2002.5, 2007.5, 2012.5, 2017.5)) %>%
    mutate(cohort = time)  %>%
    mutate(count = 1e-6 * denom_china * count)

cohorts.pmean <- population.pmean %>%
    filter(cohort >= 1932.5 & cohort <= 2007.5) %>%
    filter(time %in% c(1990, 2000, 2010)) %>%
    bind_rows(filter(births.pmean, cohort <= 2007.5)) %>%
    select(-age) %>%
    mutate(cohort.name = sprintf("%s-%s",round(cohort-1.5),round(cohort+2.5))) %>%
    arrange(cohort.name, time)

cohorts.pmean.births <- cohorts.pmean %>%
    mutate(count = ifelse(time == cohort, count, NA))

census_df <- census  %>%
    as.data.frame(direction = "long", midpoints = "age") %>%
    mutate(cohort = time - age) %>%
    mutate(count = 1e-6 * denom_china * count)

births_df <- births_un %>%
    collapseDimension(dimension = "age") %>%
    as.data.frame(direction = "long") %>%
    mutate(count = 1e-6 * denom_china * count) %>%
    mutate(time = c(1992.5, 1997.5, 2002.5, 2007.5, 2012.5)) %>%
    mutate(cohort = time)

cohorts <- census_df %>%
    filter(cohort >= 1932.5) %>%
    select(time, cohort, count) %>%
    bind_rows(filter(births_df, cohort <= 2007.5)) %>%
    mutate(cohort.name = sprintf("%s-%s",round(cohort-1.5),round(cohort+2.5))) %>%
    arrange(cohort.name,time)
cohorts.births <- cohorts %>%
    mutate(count = ifelse(time == cohort, count, NA))

p <- xyplot(count ~ time | cohort.name,
```

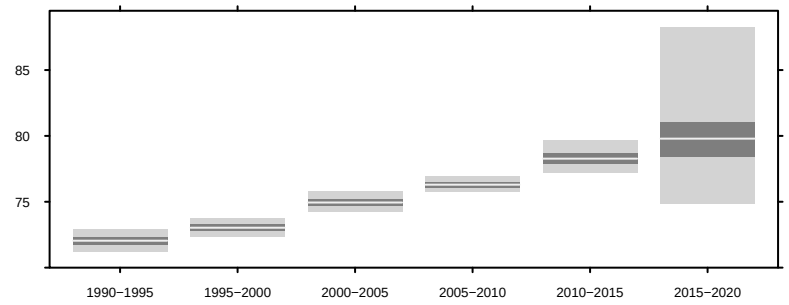
```
            data = cohorts,
            subset = cohort >= 1952.5,
            type = "b",
            col = "black",
            ylim = c(0, NA),
            xlab = "",
            ylab = "",
            xlim = c(1987, 2013),
            as.table = TRUE,
            layout = c(NA,2),
            scales = list(tck = 0.3),
            par.settings = list(strip.background = list(col = "grey90"),
                                fontsize = list(points=6, text=6)))
p <- p + as.layer(xyplot(count ~ time | cohort.name,
                         data = cohorts.births,
                         subset = cohort>=1952.5,
                         type = "p",
                         pch = 19,
                         col = "black"))
p <- p + as.layer(xyplot(count ~ time | cohort.name,
                         data = cohorts.pmean,
                         subset = cohort>=1952.5,
                         type = "b",
                         col = "grey"))
p <- p + as.layer(xyplot(count ~ time | cohort.name,
                         data = cohorts.pmean.births,
                         subset = cohort>=1952.5,
                         type = "p",
                         pch = 19,
                         col = "grey"))
p$condlevels[[1]] <- period_labels
plot(p)
```



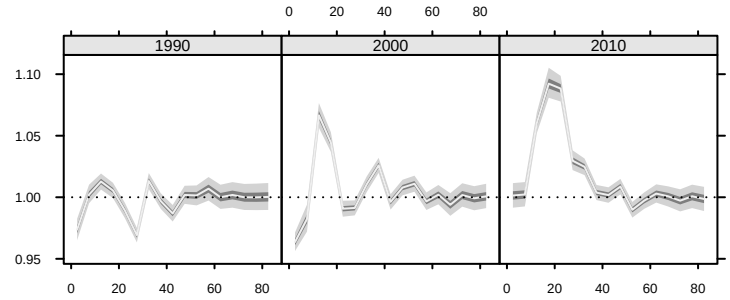

```
#Death rates
period_labels <- paste(seq(1990, 2015, 5), seq(1995, 2020, 5), sep = "-")

rates <- fetch(filename,
               where = c("systemModels", "deaths", "likelihood", "rate"))
weights <- fetch(filename,
                 where = c("account", "population")) %>%
    exposure(triangles = TRUE)

p <- dplot(~ age | factor(time),
           data = rates,
           weights = weights,
           col = "grey",
           xlab = "",
           ylab = "",
           midpoints = "age",
           as.table = TRUE,
           scales = list(tck = 0.3,
                         y = list(log = TRUE)),
           yscale.components = yscale.components.log10ticks,
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))
p$condlevels[[1]] <- period_labels

plot(p)
```




```
#Life expectancy
period_labels <- paste(seq(1990, 2015, 5), seq(1995, 2020, 5), sep = "-")

rates <- fetch(filename,
               where = c("systemModels", "deaths", "likelihood", "rate"))
weights <- fetch(filename,
                 where = c("account", "population")) %>%
    exposure(triangles = TRUE)

lifeexp <- rates %>%
    collapseDimension(dimension="triangle", weights=weights) %>%
    LifeTable() %>%
    lifeExpectancy()

lifeexp <- as(lifeexp, "array")
dimnames(lifeexp)[[1]] <- period_labels
lifeexp <- Counts(lifeexp,
                  dimtypes = c(time = "state"))

p <- dplot(~ time,
           data = lifeexp,
           col = "grey",
           xlab = "",
           ylab = "",
           as.table = TRUE,
           scales = list(tck = 0.3),
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))
plot(p)
```

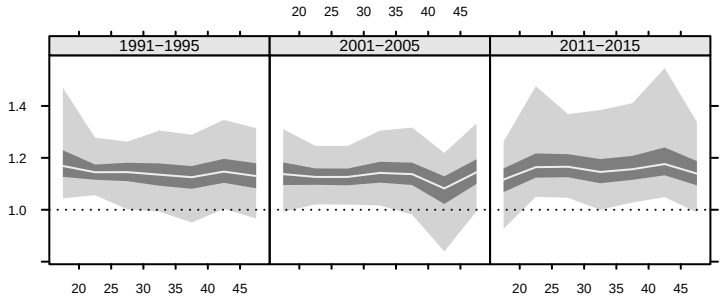



```
#Coverage ratio for census counts
coverage <- fetchCoverage(filename, "census")

p <- dplot(~ age | factor(time),
           data = coverage,
           col = "grey",
           xlab = "",
           ylab = "",
           midpoint = "age",
           as.table = TRUE,
           scales = list(x = list(tck = 0.3),
                         y = list(tck = 0.5)),
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))
p <- p + latticeExtra::layer(panel.abline(h=1,lty="dotted",col="black"))
plot(p)
```




```
#Coverage ratio for UN births estimates
coverage <- fetchCoverage(filename, "births_un") %>%
    subarray(time %in% c("1991-1995", "2001-2005", "2011-2015"))

p <- dplot(~ age | factor(time),
           data = coverage,
           col = "grey",
           xlab = "",
           ylab = "",
           midpoint = "age",
           layout = c(NA,1),
           as.table = TRUE,
           scales = list(x = list(tck = 0.3),
```

```
                          y = list(tck = 0.5)),
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))

p <- p + latticeExtra::layer(panel.abline(h = 1,lty = "dotted", col = "black"))
plot(p)
```




```
#Coverage ratio for UN deaths estimates
coverage <- fetchCoverage(filename, "deaths_un") %>%
    subarray(time %in% c("1991-1995", "2001-2005", "2011-2015"))

p <- dplot(~ age | factor(time),
           data = coverage,
           col = "grey",
           xlab = "",
           ylab = "",
           midpoint = "age",
           layout = c(NA,1),
           as.table = TRUE,
           scales = list(x = list(tck = 0.3),
                         y = list(tck = 0.5)),
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))

p <- p + latticeExtra::layer(panel.abline(h = 1, lty = "dotted", col = "black"))
plot(p)
```