Chapter 18 Population in New Zealand

John Bryant and Junni L. Zhang

2018-09-22

Setup

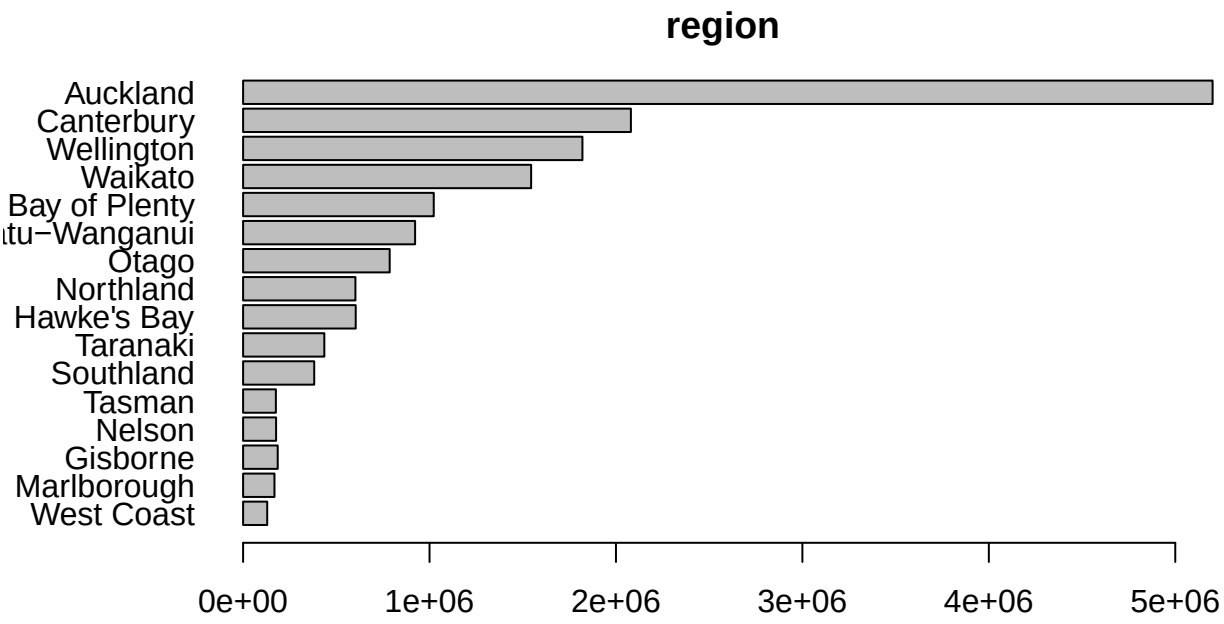
```
library(methods)
library(bdefdata)
library(demest)
library(dplyr)
library(tidyverse)
library(latticeExtra)
```

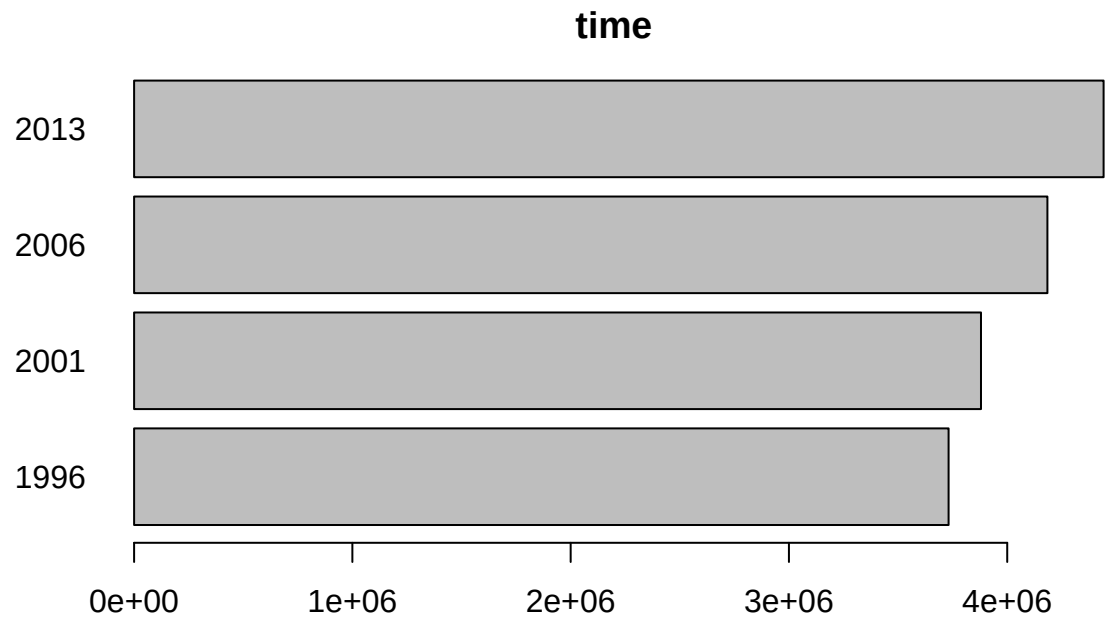
Get data

```
population_census <- bdefdata::new_zealand_popn_census %>%
    Counts()
summary(population_census)
```

```
##
## name:     region     time
## length:   16         4
## dimtype:  state      time
## dimscale: Categories Points
## first:    West Coast 1996
## last:     Auckland   2013
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   31050   46745  149720  253696  299450 1493210
```

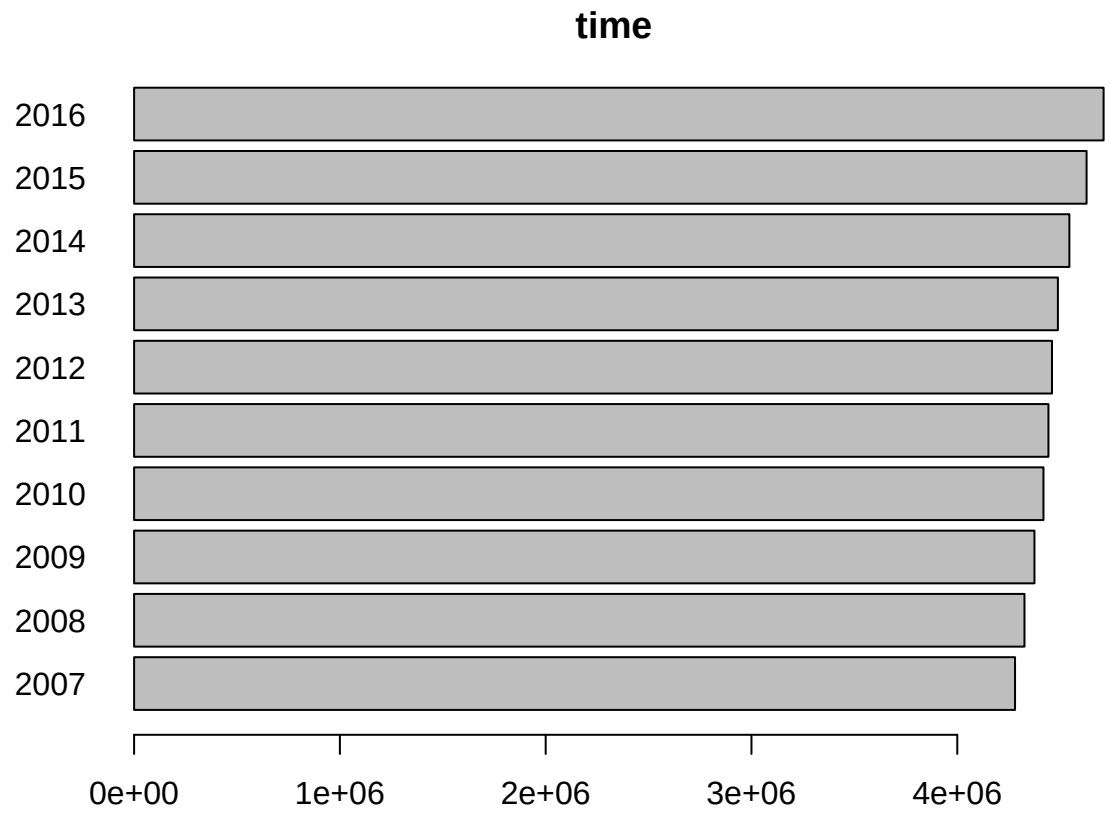
```
plot(population_census)
```






```
population_admin <- bdefdata::new_zealand_popn_admin %>%
    Counts(dimscales = c(time = "Points"))
```

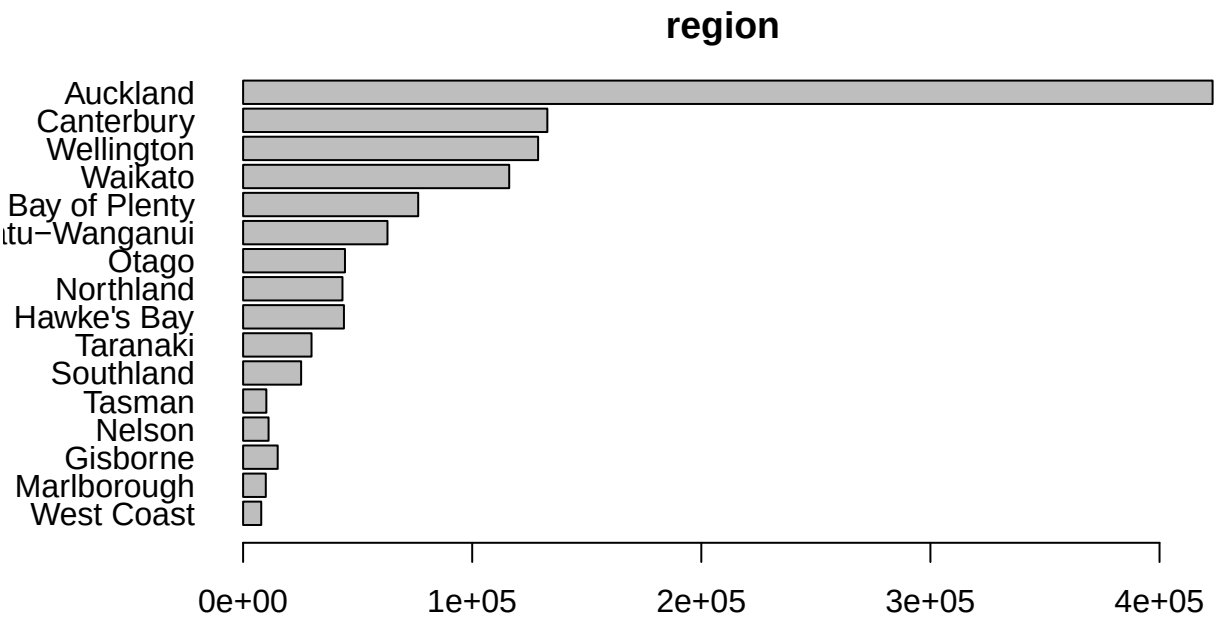
```
plot(population_admin)
```

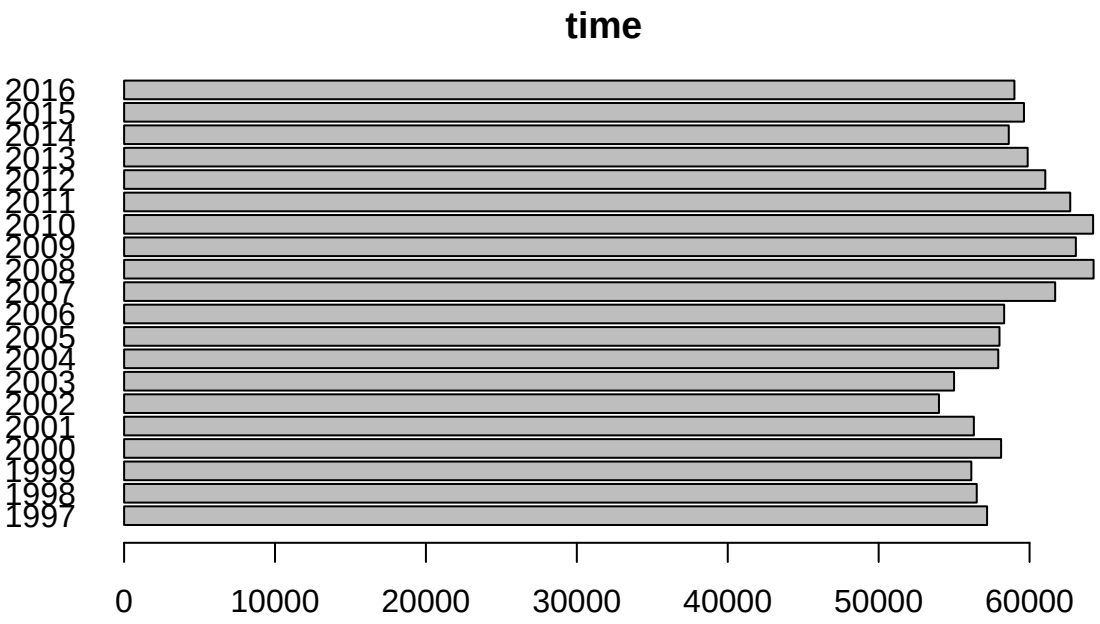



```
births_reg <- bdefdata::new_zealand_births_reg %>%
    Counts(dimscales = c(time = "Intervals"))
summary(births_reg)
```

```
##
## name:     region     time
## length:   16         20
## dimtype:  state      time
## dimscale: Categories Intervals
## first:    West Coast 1997
## last:     Auckland   2016
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   318.0   671.2  2142.0  3692.3  4407.0 23544.0
```

```
plot(births_reg)
```



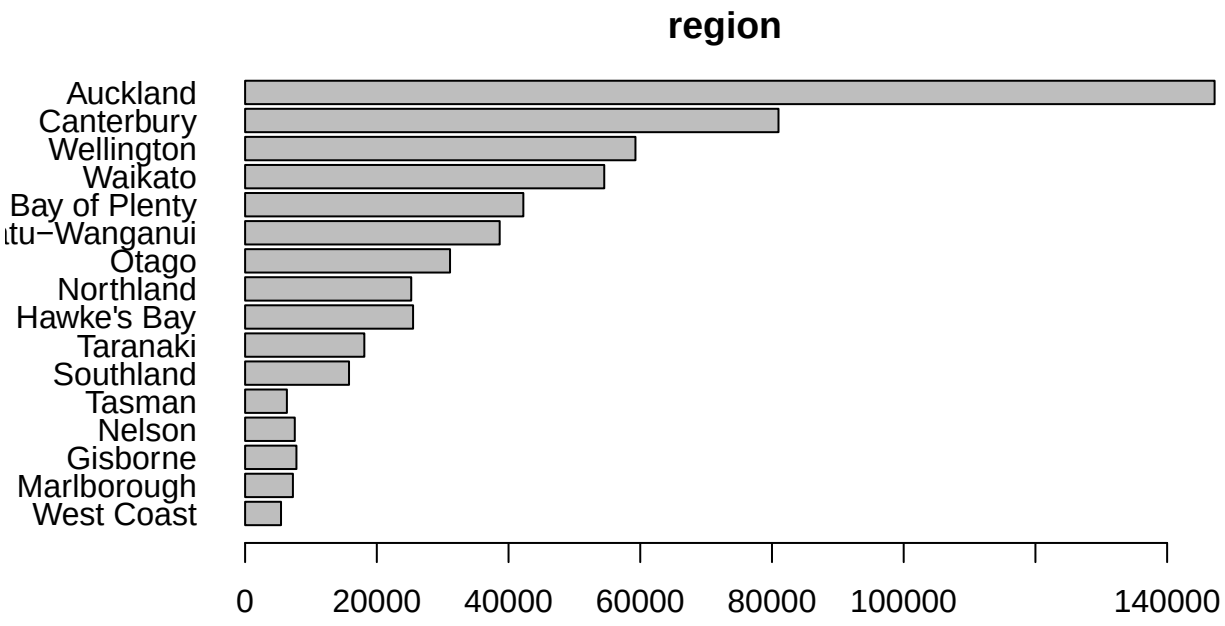


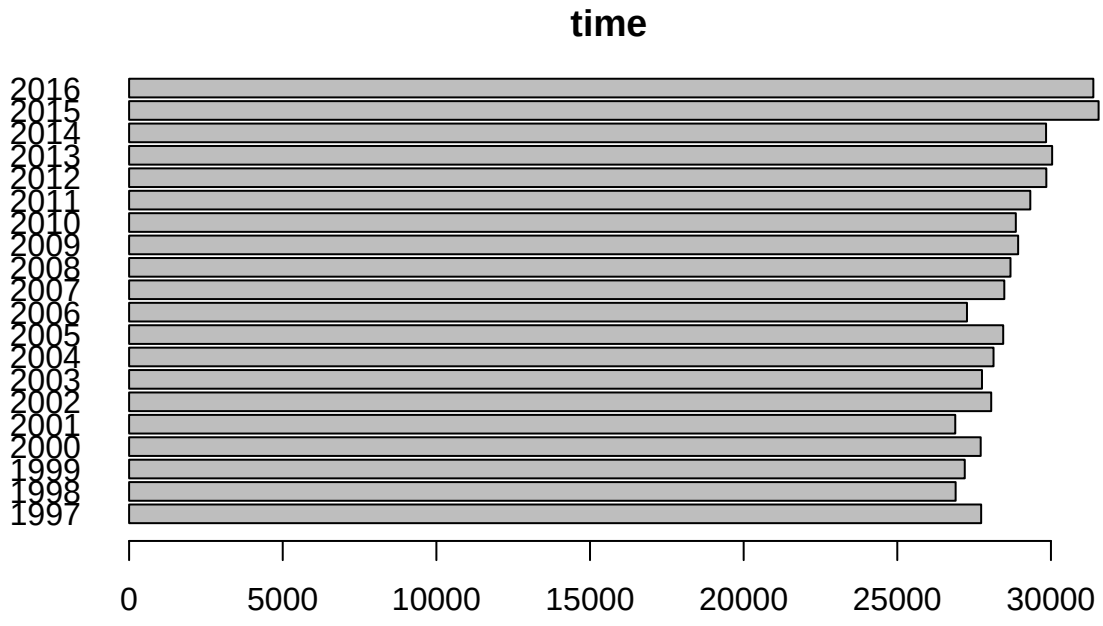

```
deaths_reg <- bdefdata::new_zealand_deaths_reg %>%
    Counts(dimscales = c(time = "Intervals"))
summary(deaths_reg)
```

```
##
## name:     region     time
## length:   16         20
## dimtype:  state      time
## dimscale: Categories Intervals
## first:    West Coast 1997
## last:     Auckland   2016
```

```
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   234.0   386.2  1261.5  1790.7  2478.8  8139.0
```

```
plot(deaths_reg)
```



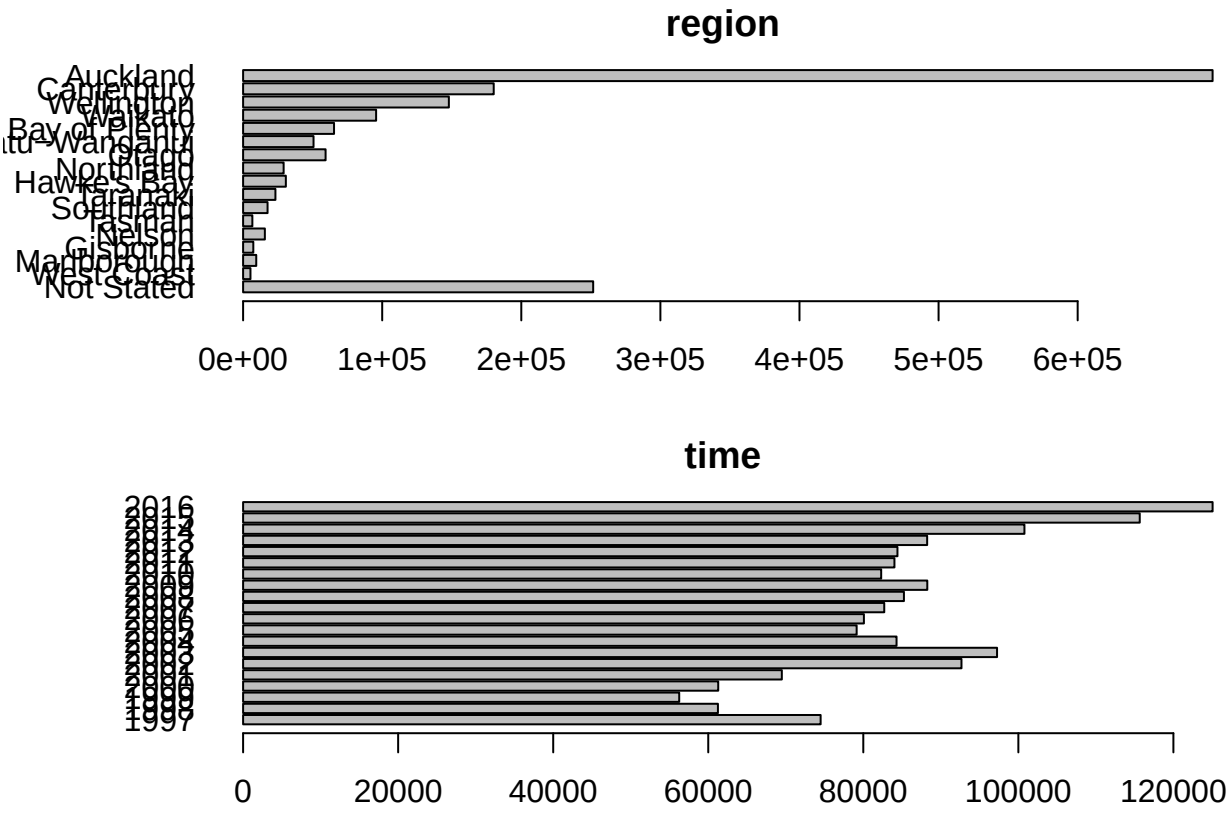



```
arrivals <- bdefdata::new_zealand_arrivals %>%
    Counts(dimscales = c(time = "Intervals"))
summary(arrivals)
```

```
##
## name:     region     time
```

```
## length:   17         20
## dimtype:  state      time
## dimscale: Categories Intervals
## first:    Not Stated 1997
## last:     Auckland   2016
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   151.0   645.5  1628.0  4978.9  5294.8 52934.0
```

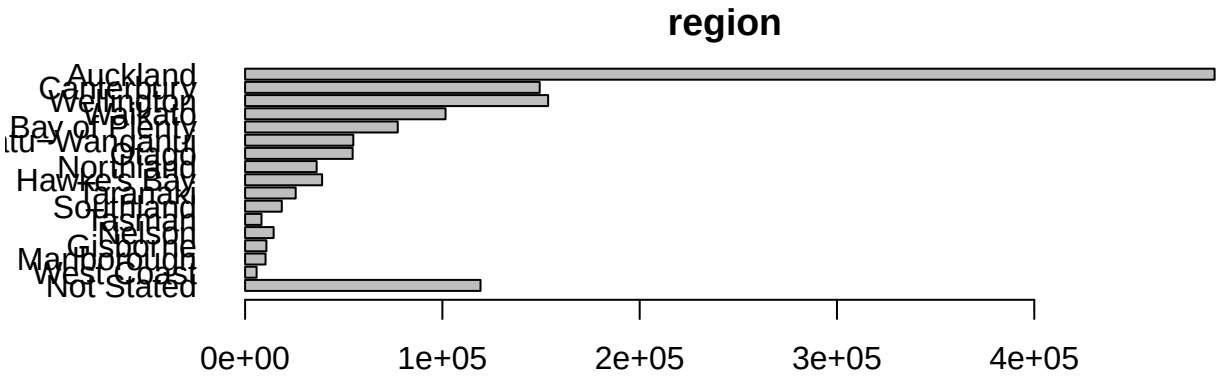
```
plot(arrivals)
```

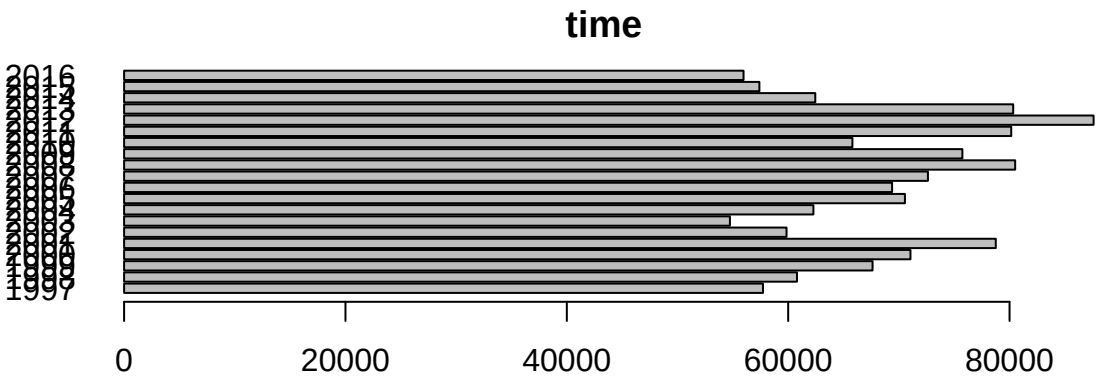



```
departures <- bdefdata::new_zealand_departures %>%
    Counts(dimscales = c(time = "Intervals"))
summary(departures)
```

```
##
## name:     region     time
## length:   17         20
## dimtype:  state      time
## dimscale: Categories Intervals
## first:    Not Stated 1997
## last:     Auckland   2016
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   212.0   703.2  1979.5  4033.2  5366.8 30269.0
```

```
plot(departures)
```

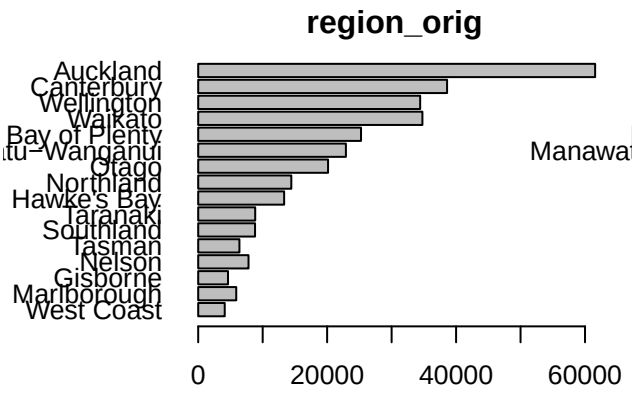


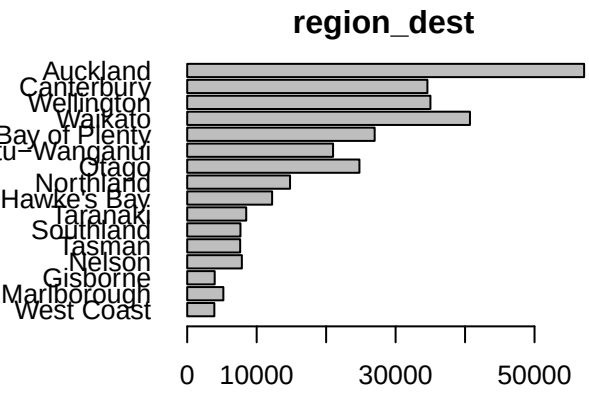


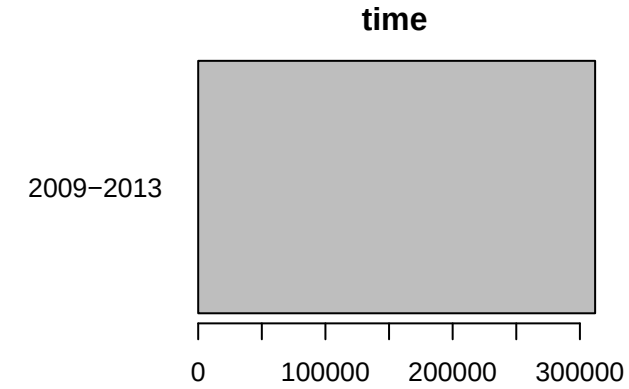

```
address_census <- bdefdata::new_zealand_address_census %>%
    Counts(dimscales = c(time = "Intervals"))
summary(address_census)
```

```
##
## name:     region_orig region_dest time
## length:   16          16          1
## dimtype:  origin      destination time
## dimscale: Categories  Categories  Intervals
## first:    West Coast  West Coast  2009-2013
## last:     Auckland    Auckland    2009-2013
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     126     381    1219    1462   15678
```

```
plot(address_census)
```



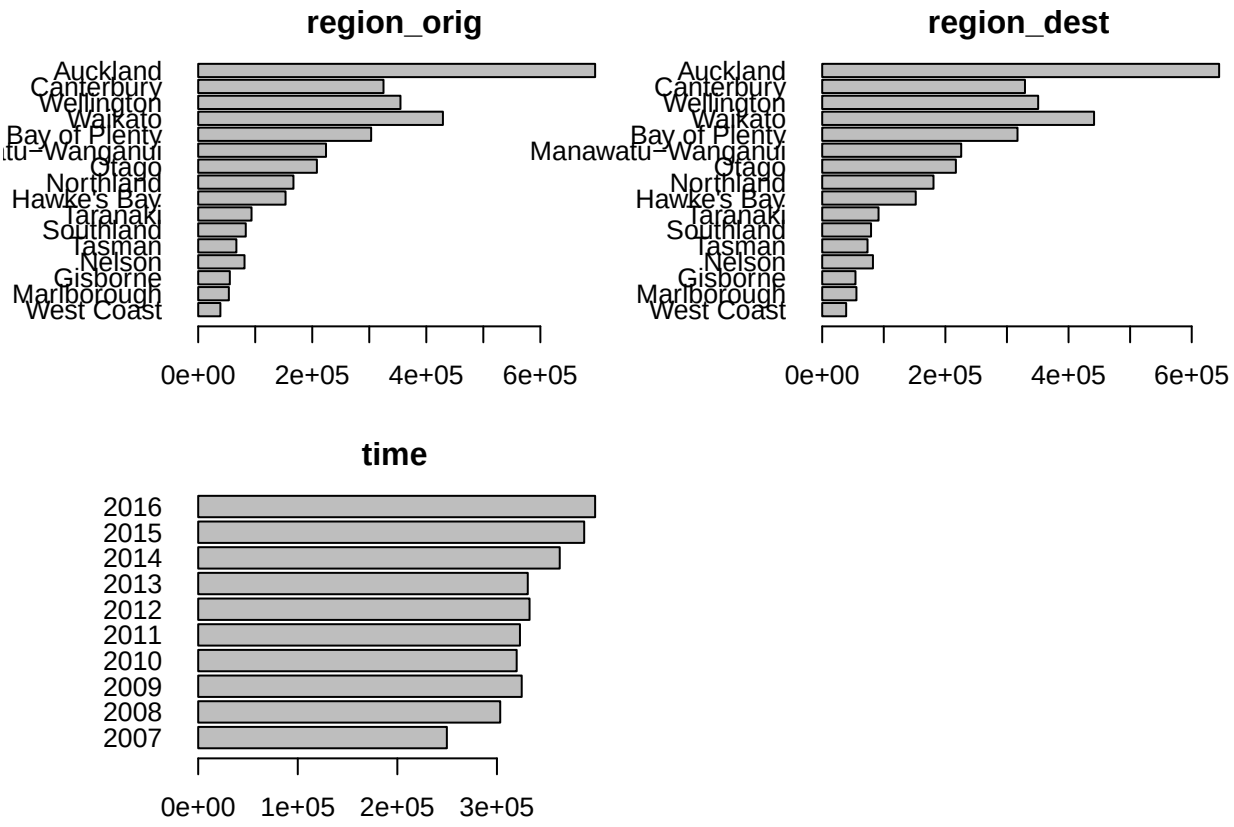





```
address_admin <- bdefdata::new_zealand_address_admin %>%
    Counts(dimscales = c(time = "Intervals"))
summary(address_admin)
```

```
##
## name:     region_orig region_dest time
## length:   16          16          10
## dimtype:  origin      destination time
## dimscale: Categories  Categories  Intervals
## first:    West Coast  West Coast  2007
## last:     Auckland    Auckland    2016
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     141     399    1302    1473   23670
```

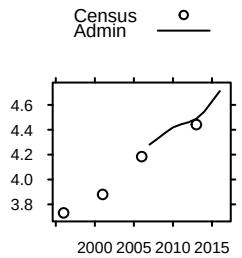
```
plot(address_admin)
```



Input Data for the National Demographic Account

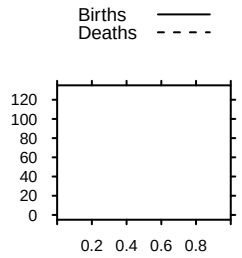
```
#Plot population estimates
population_census_df <- population_census %>%
    collapseDimension(margin = "time")  %>%
    as.data.frame(direction = "long") %>%
    mutate(variant = "Census")

population_admin_df <- population_admin %>%
    as.data.frame(direction = "long") %>%
    mutate(variant = "Admin")

population <- bind_rows(population_census_df, population_admin_df) %>%
    mutate(count = 1e-6 * count) %>%
    mutate(variant = factor(variant,
                            levels = c("Census", "Admin"),
                            labels = c("Census", "Admin")))

pch <- c(1L, NA_integer_)
lty <- c("blank", "solid")
xyplot(count ~ time,
       data = population,
       groups = variant,
       scales = list(tck = 0.3),
       par.settings = list(fontsize = list(text = 7, points = 6),
```

```
                            strip.background = list(col = "grey90")),
       type = "b",
       lty = lty,
       pch = pch,
       xlab = "",
       ylab = "",
       col = "black",
       key = list(text = list(levels(population$variant)),
                  lines = list(lty = lty,
                  pch = pch,
                  col = "black",
                  type = "b",
                  cex = 0.7),
                  divide = 1,
                  padding.text = 0.5,
                  columns = 1))
```



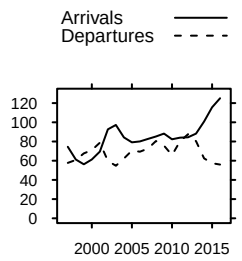

```
#Plot registered births and deaths
births_df <- births_reg %>%
    collapseDimension(margin = "time") %>%
    as.data.frame(direction = "long", midpoints = "time") %>%
    mutate(variant = "Births")

deaths_df <- deaths_reg %>%
    collapseDimension(margin = "time") %>%
    as.data.frame(direction = "long", midpoints = "time") %>%
    mutate(variant = "Deaths")

combined <- bind_rows(births_df, deaths_df) %>%
    mutate(count = 1e-3 * count) %>%
    mutate(time = as.integer(levels(time))[time]) %>%
    mutate(variant = factor(variant,
                            levels = c("Births", "Deaths"),
                            labels = c("Births", "Deaths")))


lty <- c("solid", "dashed")
xyplot(count ~ time,
       data = combined,
       groups = variant,
       scales = list(tck = 0.3),
       par.settings = list(fontsize = list(text = 7),
                           strip.background = list(col = "grey90")),
       type = "l",
```

```
       lty = lty,
       xlab = "",
       ylab = "",
       ylim = c(-5, 135),
       col = "black",
       key = list(text = list(levels(combined$variant)),
                  lines = list(lty = lty,
                  col = "black"),
                  columns = 1))
```




```
#Plot international arrivals and departures
arrivals_df <- arrivals %>%
    collapseDimension(margin = "time") %>%
    as.data.frame(direction = "long") %>%
    mutate(variant = "Arrivals")

departures_df <- departures %>%
    collapseDimension(margin = "time") %>%
    as.data.frame(direction = "long") %>%
    mutate(variant = "Departures")


combined <- bind_rows(arrivals_df, departures_df) %>%
    mutate(count = 1e-3 * count) %>%
    mutate(time = as.integer(levels(time))[time]) %>%
    mutate(variant = factor(variant,
                            levels = c("Arrivals", "Departures"),
                            labels = c("Arrivals", "Departures")))

lty <- c("solid", "dashed")
xyplot(count ~ time,
       data = combined,
       groups = variant,
       scales = list(tck = 0.3),
       par.settings = list(fontsize = list(text = 7),
                           strip.background = list(col = "grey90")),
       type = "l",
       lty = lty,
       xlab = "",
       ylab = "",
       ylim = c(-5, 135),
       col = "black",
       key = list(text = list(levels(combined$variant)),
                  lines = list(lty = lty,
```

```
                                col = "black"),
                columns = 1))
```



Fit model for National Demographic Account

```
filename <- "new_zealand_nation.est"
if(!file.exists(filename)) {
    population_census_nation <- population_census %>%
        collapseDimension(margin = "time")
    population_admin_nation <- population_admin %>%
        collapseDimension(margin = "time")
    births_reg_nation <- births_reg %>%
        collapseDimension(margin = "time")
    deaths_reg_nation <- deaths_reg %>%
        collapseDimension(margin = "time")
    arrivals_nation <- arrivals %>%
        collapseDimension(margin = "time")
    departures_nation <- departures %>%
        collapseDimension(margin = "time")

    #Initial account
    population_nation <- extrapolate(population_admin_nation,
                             labels = 1996:2006,
                             growth = growth(population_admin_nation, type = "linear"),
                             type = "linear") %>%
        toInteger(force = TRUE)
    account <- Movements(population = population_nation,
                         births = births_reg_nation,
                         entries = list(external_in = arrivals_nation),
                         exits = list(deaths = deaths_reg_nation,
                                      external_out = departures_nation)) %>%
        makeConsistent()

    #Set up system models
    systemModels <- list(Model(population ~ Poisson(mean ~ time, useExpose = FALSE),
                               time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                                          trend = Trend(scale = HalfT(scale = 0.05)),
                                          damp = NULL,
                                          error = Error(scale = HalfT(scale = 0.05))),
                               jump = 0.0015),
                         Model(births ~ Poisson(mean ~ time),
                               time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
```

```
                                        trend = NULL,
                                        damp = NULL,
                                        error = Error(scale = HalfT(scale = 0.05))),
                                  jump = 0.01),
                           Model(deaths ~ Poisson(mean ~ time),
                                  time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                                        trend = NULL,
                                        damp = NULL,
                                        error = Error(scale = HalfT(scale = 0.05))),
                                  jump = 0.01),
                           Model(external_in ~ Poisson(mean ~ time),
                                  time ~ DLM(level = Level(scale = HalfT(scale = 0.2)),
                                        trend = NULL,
                                        damp = NULL,
                                        error = Error(scale = HalfT(scale = 0.2))),
                                  jump = 0.01),
                           Model(external_out ~ Poisson(mean ~ time),
                                  time ~ DLM(level = Level(scale = HalfT(scale = 0.2)),
                                        trend = NULL,
                                        damp = NULL,
                                        error = Error(scale = HalfT(scale = 0.2))),
                                  jump = 0.01))


#Get datasets.
datasets <- list(population_census_nation = population_census_nation,
                 population_admin_nation = population_admin_nation,
                 births_reg_nation = births_reg_nation,
                 deaths_reg_nation = deaths_reg_nation,
                 arrivals_nation = arrivals_nation,
                 departures_nation = departures_nation)

#Set up data models
sd <- population_census_nation %>%
    collapseDimension(margin = "time") %>%
    as("Values")
sd <- sd * 0.0025
mean <- sd / sd

dataModels <- list(Model(population_census_nation ~ NormalFixed(mean = mean, sd = sd),
                         series = "population"),
                   Model(population_admin_nation ~ Poisson(mean ~ 1),
                         `(Intercept)` ~ ExchFixed(sd = 0.025),
                         series = "population",
                         priorSD = HalfT(scale = 0.025),
                         jump = 0.001),
                   Model(births_reg_nation ~ PoissonBinomial(prob = 0.98),
                         series = "births"),
                   Model(deaths_reg_nation ~ PoissonBinomial(prob = 0.99),
                         series = "deaths"),
                   Model(arrivals_nation ~ Poisson(mean ~ 1),
                         `(Intercept)` ~ ExchFixed(sd = 0.025),
                         lower = 0.8,
```

```
                            upper = 1.2,
                            priorSD = HalfT(scale = 0.025),
                            series = "external_in",
                            jump = 0.01),
                      Model(departures_nation ~ Poisson(mean ~ 1),
                            `(Intercept)` ~ ExchFixed(sd = 0.025),
                            lower = 0.8,
                            upper = 1.2,
                            priorSD = HalfT(scale = 0.025),
                            series = "external_out",
                            jump = 0.01))

    #Estimate account.
    n_burnin <- 500000
    n_sim <- 500000
    n_chain <- 4
    n_thin <- 1000

    set.seed(0)

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

```
## ------------------------------------------------
## Account:
## *population*
## dimensions: time
## n cells: 21
## *births*
## dimensions: time
## n cells: 20
## *external_in*
## dimensions: time
## n cells: 20
## *deaths*
## dimensions: time
## n cells: 20
## *external_out*
## dimensions: time
## n cells: 20
## ------------------------------------------------
## System models:
## *population*
## population ~ Poisson(mean ~ time, useExpose = FALSE),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = Trend(scale = HalfT(scale = 0.05)),
```

```
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.0015
## dimensions: time
## *births*
## births ~ Poisson(mean ~ time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.01
## dimensions: time
## *external_in*
## external_in ~ Poisson(mean ~ time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.2)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.2))),
## 0.01
## dimensions: time
## *deaths*
## deaths ~ Poisson(mean ~ time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.01
## dimensions: time
## *external_out*
## external_out ~ Poisson(mean ~ time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.2)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.2))),
## 0.01
## dimensions: time
## ------------------------------------------------
## Datasets:
## *population_census_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 4,  n missing: 0, integers: TRUE, n zeros: 0, median: 4031865
## *population_admin_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 10,  n missing: 0, integers: TRUE, n zeros: 0, median: 4451648
## *births_reg_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 20,  n missing: 0, integers: TRUE, n zeros: 0, median: 58467
## *deaths_reg_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 20,  n missing: 0, integers: TRUE, n zeros: 0, median: 28464
## *arrivals_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 20,  n missing: 0, integers: TRUE, n zeros: 0, median: 84150.5
## *departures_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 20,  n missing: 0, integers: TRUE, n zeros: 0, median: 68504.5
## ------------------------------------------------
```

```
## Data models:
## *population_census_nation*
## population_census_nation ~ NormalFixed(mean = mean, sd = sd),
## "population"
## dimensions: time
## *population_admin_nation*
## population_admin_nation ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.025),
## HalfT(scale = 0.025),
## 0.001,
## "population"
## dimensions: time
## *births_reg_nation*
## births_reg_nation ~ PoissonBinomial(prob = 0.98),
## "births"
## dimensions: time
## *deaths_reg_nation*
## deaths_reg_nation ~ PoissonBinomial(prob = 0.99),
## "deaths"
## dimensions: time
## *arrivals_nation*
## arrivals_nation ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.025),
## 0.8,
## 1.2,
## HalfT(scale = 0.025),
## 0.01,
## "external_in"
## dimensions: time
## *departures_nation*
## departures_nation ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.025),
## 0.8,
## 1.2,
## HalfT(scale = 0.025),
## 0.01,
## "external_out"
## dimensions: time
## ------------------------------------------------
## MCMC statistics:
## nBurnin: 500000,  nSim: 500000,  nChain: 4,  nThin: 1000,  nIteration: 2000
##
## Metropolis-Hastings updates:
##                                                    jump acceptance
## systemModels.population.likelihood.count          0.002      0.349
## systemModels.births.likelihood.rate               0.010      0.427
## systemModels.external_in.likelihood.rate          0.010      0.382
## systemModels.deaths.likelihood.rate               0.010      0.531
## systemModels.external_out.likelihood.rate         0.010      0.411
## dataModels.population_admin_nation.likelihood.rate 0.001      0.371
## dataModels.arrivals_nation.likelihood.rate        0.010      0.377
## dataModels.departures_nation.likelihood.rate      0.010      0.399
##                                                   autocorr
## systemModels.population.likelihood.count             0.512
```

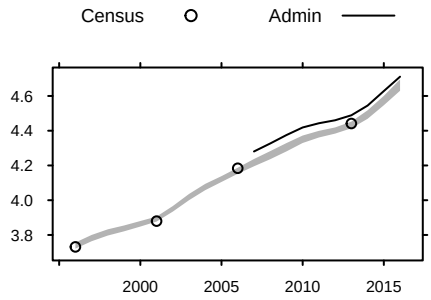
```
## systemModels.births.likelihood.rate                   0.119
## systemModels.external_in.likelihood.rate              0.653
## systemModels.deaths.likelihood.rate                   0.079
## systemModels.external_out.likelihood.rate             0.627
## dataModels.population_admin_nation.likelihood.rate    0.678
## dataModels.arrivals_nation.likelihood.rate            0.652
## dataModels.departures_nation.likelihood.rate          0.629
##
## parameters:
##                                                      Rhat      2.5%
## account.population                                   1.05   3737590
## account.births                                       1.01     54012
## account.external_in                                . 1.12     59934
## account.deaths                                       1.00     26881
## account.external_out                                 1.08     54014
## systemModels.population.likelihood.count             1.05   3737976
## systemModels.population.prior.mean                   1.05      15.1
## systemModels.population.hyper.time.scaleLevel        1.00  0.000173
## systemModels.population.hyper.time.scaleTrend        1.00  0.000363
## systemModels.population.hyper.time.scaleError        1.01  0.000661
## systemModels.births.likelihood.rate                  1.01    0.0128
## systemModels.births.prior.mean                       1.01     -4.35
## systemModels.births.hyper.time.scaleLevel            1.00    0.0153
## systemModels.births.hyper.time.scaleError            1.00   0.00678
## systemModels.external_in.likelihood.rate           . 1.12    0.0156
## systemModels.external_in.prior.mean                  1.03     -4.14
## systemModels.external_in.hyper.time.scaleLevel       1.00    0.0405
## systemModels.external_in.hyper.time.scaleError       1.02    0.0419
## systemModels.deaths.likelihood.rate                  1.01   0.00661
## systemModels.deaths.prior.mean                       1.01     -5.01
## systemModels.deaths.hyper.time.scaleLevel            1.00   0.00542
## systemModels.deaths.hyper.time.scaleError            1.02   0.00646
## systemModels.external_out.likelihood.rate            1.07    0.0124
## systemModels.external_out.prior.mean                 1.03     -4.37
## systemModels.external_out.hyper.time.scaleLevel      1.00   0.00705
## systemModels.external_out.hyper.time.scaleError      1.00    0.0375
## dataModels.population_admin_nation.likelihood.rate   1.05      1.01
## dataModels.population_admin_nation.prior.mean        1.00   0.00832
## dataModels.population_admin_nation.prior.sd          1.06 0.0000127
## dataModels.arrivals_nation.likelihood.rate         . 1.12     0.848
## dataModels.arrivals_nation.prior.mean                1.01   -0.0325
## dataModels.arrivals_nation.prior.sd                . 1.17   0.00432
## dataModels.departures_nation.likelihood.rate         1.07     0.868
## dataModels.departures_nation.prior.mean              1.03   -0.0373
## dataModels.departures_nation.prior.sd              . 1.12   0.00535
##                                                        50%   97.5% length
## account.population                                 4170990 4649138     21
## account.births                                       58489   64266     20
## account.external_in                                  84046  115671     20
## account.deaths                                       28458   31555     20
## account.external_out                                 67254   91679     20
## systemModels.population.likelihood.count           4171783 4649255     21
## systemModels.population.prior.mean                    15.2    15.4     21
## systemModels.population.hyper.time.scaleLevel      0.00169 0.00566      1
```

```
## systemModels.population.hyper.time.scaleTrend       0.00296 0.00521      1
## systemModels.population.hyper.time.scaleError       0.00167  0.0286      1
## systemModels.births.likelihood.rate                  0.0143  0.0152     20
## systemModels.births.prior.mean                        -4.24   -4.18     20
## systemModels.births.hyper.time.scaleLevel            0.0258  0.0391      1
## systemModels.births.hyper.time.scaleError            0.0159  0.0466      1
## systemModels.external_in.likelihood.rate             0.0199  0.0257     20
## systemModels.external_in.prior.mean                   -3.91   -3.66     20
## systemModels.external_in.hyper.time.scaleLevel       0.0848   0.142      1
## systemModels.external_in.hyper.time.scaleError       0.0848    0.18      1
## systemModels.deaths.likelihood.rate                 0.00686 0.00733     20
## systemModels.deaths.prior.mean                        -4.98   -4.93     20
## systemModels.deaths.hyper.time.scaleLevel             0.012  0.0246      1
## systemModels.deaths.hyper.time.scaleError            0.0156  0.0224      1
## systemModels.external_out.likelihood.rate            0.0164  0.0209     20
## systemModels.external_out.prior.mean                  -4.12    -3.9     20
## systemModels.external_out.hyper.time.scaleLevel      0.0908   0.178      1
## systemModels.external_out.hyper.time.scaleError      0.0895    0.57      1
## dataModels.population_admin_nation.likelihood.rate     1.01    1.02     10
## dataModels.population_admin_nation.prior.mean        0.0137  0.0173     10
## dataModels.population_admin_nation.prior.sd        0.000619 0.00603      1
## dataModels.arrivals_nation.likelihood.rate             1.01    1.14     20
## dataModels.arrivals_nation.prior.mean               0.00169  0.0406     20
## dataModels.arrivals_nation.prior.sd                  0.0649   0.112      1
## dataModels.departures_nation.likelihood.rate          0.991    1.15     20
## dataModels.departures_nation.prior.mean             -0.0106  0.0552     20
## dataModels.departures_nation.prior.sd                0.0586   0.118      1
## -------------------------------------------------
```

Results for the National Demographic Account

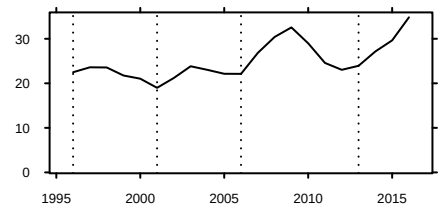
```
#Population estimates
population <- fetch(filename, where = c("account", "population"))

direct <- bind_rows(population_census_df, population_admin_df) %>%
    mutate(count = 1e-6 * count) %>%
    mutate(variant = factor(variant,
                            levels = c("Census", "Admin")))

pch <- c(1L, NA_integer_)
lty <- c("blank", "solid")
p <- dplot(~ time,
           data = 1e-6 * population,
           col = "grey70",
           xlab = "",
           ylab = "",
           as.table = TRUE,
           scales = list(tck = 0.3),
           prob = c(0.025, 0.975),
           par.settings = list(fontsize = list(text = 7),
                               strip.background = list(col = "grey90")),
           key = list(text = list(levels(direct$variant)),
                      lines = list(lty = lty,
```

```
                                    pch = pch,
                                    col = "black",
                                    type = "b",
                                    cex = 0.7),
                        divide = 1,
                        padding.text = 0.5,
                        columns = 2))
p <- p + as.layer(xyplot(count ~ time,
                         data = direct,
                         groups = variant,
                         scales = list(tck = 0.3),
                         par.settings = list(fontsize = list(text = 7, points = 6),
                                             strip.background = list(col = "grey90")),
                         type = "b",
                         lty = lty,
                         pch = pch,
                         xlab = "",
                         ylab = "",
                         col = "black"))
plot(p)
```



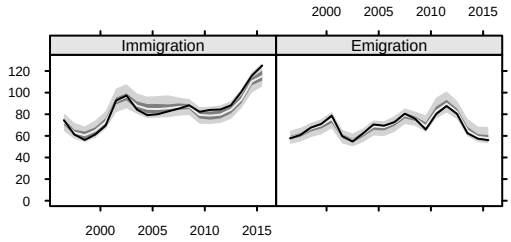

```
#Widths of 95% credible intervals for population estimates
##NEED TO CHECK CODE, COMPARE WITH THE GRAPH IN THE BOOK
popn_cred_int <- fetch(filename, where = c("account", "population")) %>%
    collapseIterations(prob = c(0.1, 0.9))

widths <- slab(popn_cred_int, dimension = "quantile", element = 2) -
    slab(popn_cred_int, dimension = "quantile", element = 1)

p <- dplot(~ time,
           data = 1e-3 * widths,
           xlab = "",
           ylab = "",
           as.table = TRUE,
           col = "black",
           ylim = c(-0.2, NA),
           scales = list(tck = 0.3),
           lattice.options = modifyList(lattice.options(),
                                        list(skip.boundary.labels = 0)),
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))
p <- p + latticeExtra::layer(panel.abline(v = c(1996, 2001, 2006, 2013),
```

```
                                        lty = "dotted",
                                        col = "black",
                                        under = TRUE))
plot(p)
```

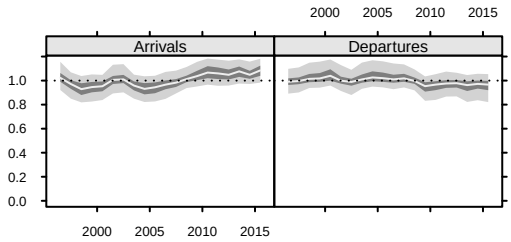



```
#Estimates of international migration
Immigration <- fetch(filename, where = c("account", "external_in"))
Emigration <- fetch(filename, where = c("account", "external_out"))

components <- dbind(Immigration,
                    Emigration,
                    along = "component")

arrivals_df <- arrivals %>%
    collapseDimension(margin = "time") %>%
    as.data.frame(direction = "long", midpoints = "time") %>%
    mutate(variant = "Arrivals")

departures_df <- departures %>%
    collapseDimension(margin = "time") %>%
    as.data.frame(direction = "long", midpoints = "time") %>%
    mutate(variant = "Departures")


combined <- bind_rows(arrivals_df, departures_df) %>%
    mutate(count = 1e-3 * count) %>%
    mutate(variant = factor(variant,
                            levels = c("Arrivals", "Departures"),
                            labels = c("Arrivals", "Departures")))

p <- dplot(~ time | component,
           data = 1e-3 * components,
           col = "grey",
           xlab = "",
           ylab = "",
           as.table = TRUE,
           scales = list(tck = 0.3),
           ylim = c(-5, 135),
           midpoints = "time",
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))
p <- p + as.layer(xyplot(count ~ time | variant,
                         data = combined,
                         scales = list(tck = 0.3),
```

```
                        par.settings = list(fontsize = list(text = 7),
                                            strip.background = list(col = "grey90")),
                        type = "l",
                        xlab = "",
                        ylab = "",
                        col = "black"))
plot(p)
```




```
#Coverage ratios for international migration
coverage <- dbind("Arrivals" = fetchCoverage(filename, dataset = "arrivals_nation"),
                  "Departures" = fetchCoverage(filename, dataset = "departures_nation"),
                  along = "dataset")

p <- dplot(~ time | dataset,
           data = coverage,
           col = "grey",
           xlab = "",
           ylab = "",
           midpoints = "time",
           ylim = c(-0.05, NA),
           as.table = TRUE,
           scales = list(tck = 0.3),
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))
p <- p + latticeExtra::layer(panel.abline(h = 1,
                                          col = "black",
                                          lty = "dotted"))
plot(p)
```



Fit an Alternative Model for Sensitivity Tests for the National Demographic Account

```
filename_sens <- "new_zealand_nation_sens.est"
if(!file.exists(filename_sens)) {
    population_census_nation <- population_census %>%
        collapseDimension(margin = "time")
    population_admin_nation <- population_admin %>%
        collapseDimension(margin = "time")
    births_reg_nation <- births_reg %>%
        collapseDimension(margin = "time")
    deaths_reg_nation <- deaths_reg %>%
        collapseDimension(margin = "time")
    arrivals_nation <- arrivals %>%
        collapseDimension(margin = "time")
    departures_nation <- departures %>%
        collapseDimension(margin = "time")

    #Initial account
    population_nation <- extrapolate(population_admin_nation,
                              labels = 1996:2006,
                              growth = growth(population_admin_nation, type = "linear"),
                              type = "linear") %>%
        toInteger(force = TRUE)
    account <- Movements(population = population_nation,
                         births = births_reg_nation,
                         entries = list(external_in = arrivals_nation),
                         exits = list(deaths = deaths_reg_nation,
                                      external_out = departures_nation)) %>%
        makeConsistent()

    #Set up system models
    systemModels <- list(Model(population ~ Poisson(mean ~ time, useExpose = FALSE),
                               time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                                          trend = Trend(scale = HalfT(scale = 0.05)),
                                          damp = NULL,
                                          error = Error(scale = HalfT(scale = 0.05))),
                               jump = 0.0015),
                         Model(births ~ Poisson(mean ~ time),
                               time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                                          trend = NULL,
                                          damp = NULL,
                                          error = Error(scale = HalfT(scale = 0.05))),
                               jump = 0.01),
                         Model(deaths ~ Poisson(mean ~ time),
                               time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                                          trend = NULL,
                                          damp = NULL,
                                          error = Error(scale = HalfT(scale = 0.05))),
                               jump = 0.01),
                         Model(external_in ~ Poisson(mean ~ time),
                               time ~ DLM(level = Level(scale = HalfT(scale = 0.2)),
                                          trend = NULL,
                                          damp = NULL,
```

```
                                        error = Error(scale = HalfT(scale = 0.2))),
                              jump = 0.01),
                        Model(external_out ~ Poisson(mean ~ time),
                              time ~ DLM(level = Level(scale = HalfT(scale = 0.2)),
                                         trend = NULL,
                                         damp = NULL,
                                         error = Error(scale = HalfT(scale = 0.2))),
                              jump = 0.01))


#Get datasets.
datasets <- list(population_census_nation = population_census_nation,
                 population_admin_nation = population_admin_nation,
                 births_reg_nation = births_reg_nation,
                 deaths_reg_nation = deaths_reg_nation,
                 arrivals_nation = arrivals_nation,
                 departures_nation = departures_nation)

#Set up data models
sd <- population_census_nation %>%
    collapseDimension(margin = "time") %>%
    as("Values")
sd <- sd * 0.0025
mean <- sd / sd

dataModels <- list(Model(population_census_nation ~ NormalFixed(mean = mean, sd = sd*2),
                         series = "population"),
                   Model(population_admin_nation ~ Poisson(mean ~ 1),
                         `(Intercept)` ~ ExchFixed(sd = 0.025),
                         series = "population",
                         priorSD = HalfT(scale = 0.025),
                         jump = 0.001),
                   Model(births_reg_nation ~ PoissonBinomial(prob = 0.98),
                         series = "births"),
                   Model(deaths_reg_nation ~ PoissonBinomial(prob = 0.99),
                         series = "deaths"),
                   Model(arrivals_nation ~ Poisson(mean ~ 1),
                         `(Intercept)` ~ ExchFixed(sd = 0.025),
                         lower = 0.8,
                         upper = 1.2,
                         priorSD = HalfT(scale = 0.025),
                         series = "external_in",
                         jump = 0.01),
                   Model(departures_nation ~ Poisson(mean ~ 1),
                         `(Intercept)` ~ ExchFixed(sd = 0.025),
                         lower = 0.8,
                         upper = 1.2,
                         priorSD = HalfT(scale = 0.025),
                         series = "external_out",
                         jump = 0.01))

#Estimate account.
n_burnin <- 500000
```

```
    n_sim <- 500000
    n_chain <- 4
    n_thin <- 1000

    set.seed(0)

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

```
## ------------------------------------------------
## Account:
## *population*
## dimensions: time
## n cells: 21
## *births*
## dimensions: time
## n cells: 20
## *external_in*
## dimensions: time
## n cells: 20
## *deaths*
## dimensions: time
## n cells: 20
## *external_out*
## dimensions: time
## n cells: 20
## ------------------------------------------------
## System models:
## *population*
## population ~ Poisson(mean ~ time, useExpose = FALSE),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = Trend(scale = HalfT(scale = 0.05)),
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.0015
## dimensions: time
## *births*
## births ~ Poisson(mean ~ time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.01
## dimensions: time
## *external_in*
## external_in ~ Poisson(mean ~ time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.2)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.2))),
## 0.01
```

```
## dimensions: time
## *deaths*
## deaths ~ Poisson(mean ~ time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.01
## dimensions: time
## *external_out*
## external_out ~ Poisson(mean ~ time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.2)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.2))),
## 0.01
## dimensions: time
## ------------------------------------------------
## Datasets:
## *population_census_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 4,  n missing: 0, integers: TRUE, n zeros: 0, median: 4031865
## *population_admin_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 10,  n missing: 0, integers: TRUE, n zeros: 0, median: 4451648
## *births_reg_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 20,  n missing: 0, integers: TRUE, n zeros: 0, median: 58467
## *deaths_reg_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 20,  n missing: 0, integers: TRUE, n zeros: 0, median: 28464
## *arrivals_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 20,  n missing: 0, integers: TRUE, n zeros: 0, median: 84150.5
## *departures_nation*
## Object of class "Counts"
## dimensions: time
## n cells: 20,  n missing: 0, integers: TRUE, n zeros: 0, median: 68504.5
## ------------------------------------------------
## Data models:
## *population_census_nation*
## population_census_nation ~ NormalFixed(mean = mean, sd = sd *
##     2),
## "population"
## dimensions: time
## *population_admin_nation*
## population_admin_nation ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.025),
## HalfT(scale = 0.025),
## 0.001,
## "population"
## dimensions: time
## *births_reg_nation*
```

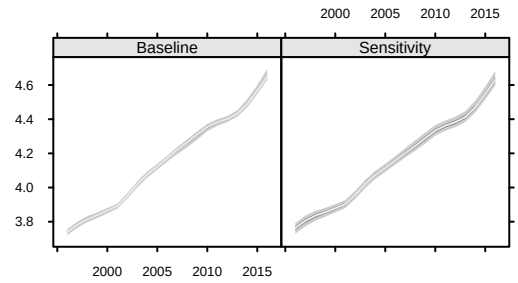
```
## births_reg_nation ~ PoissonBinomial(prob = 0.98),
## "births"
## dimensions: time
## *deaths_reg_nation*
## deaths_reg_nation ~ PoissonBinomial(prob = 0.99),
## "deaths"
## dimensions: time
## *arrivals_nation*
## arrivals_nation ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.025),
## 0.8,
## 1.2,
## HalfT(scale = 0.025),
## 0.01,
## "external_in"
## dimensions: time
## *departures_nation*
## departures_nation ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.025),
## 0.8,
## 1.2,
## HalfT(scale = 0.025),
## 0.01,
## "external_out"
## dimensions: time
## ------------------------------------------------
## MCMC statistics:
## nBurnin: 500000,  nSim: 500000,  nChain: 4,  nThin: 1000,  nIteration: 2000
##
## Metropolis-Hastings updates:
##                                                  jump acceptance
## systemModels.population.likelihood.count        0.002      0.349
## systemModels.births.likelihood.rate             0.010      0.428
## systemModels.external_in.likelihood.rate        0.010      0.379
## systemModels.deaths.likelihood.rate             0.010      0.532
## systemModels.external_out.likelihood.rate       0.010      0.415
## dataModels.population_admin_nation.likelihood.rate 0.001   0.387
## dataModels.arrivals_nation.likelihood.rate      0.010      0.371
## dataModels.departures_nation.likelihood.rate    0.010      0.403
##                                                 autocorr
## systemModels.population.likelihood.count           0.709
## systemModels.births.likelihood.rate                0.292
## systemModels.external_in.likelihood.rate           0.632
## systemModels.deaths.likelihood.rate                0.190
## systemModels.external_out.likelihood.rate          0.600
## dataModels.population_admin_nation.likelihood.rate 0.817
## dataModels.arrivals_nation.likelihood.rate         0.632
## dataModels.departures_nation.likelihood.rate       0.599
##
## parameters:
##                                                   Rhat      2.5%
## account.population                                1.03   3766381
## account.births                                    1.00     54004
## account.external_in                               1.05     57107
```

```
## account.deaths                                          1.01     26875
## account.external_out                                    1.08     55036
## systemModels.population.likelihood.count                1.03   3765789
## systemModels.population.prior.mean                      1.08      15.1
## systemModels.population.hyper.time.scaleLevel           1.00  0.000139
## systemModels.population.hyper.time.scaleTrend           1.00  0.000853
## systemModels.population.hyper.time.scaleError           1.03   0.00047
## systemModels.births.likelihood.rate                     1.02    0.0129
## systemModels.births.prior.mean                          1.01     -4.34
## systemModels.births.hyper.time.scaleLevel               1.00     0.014
## systemModels.births.hyper.time.scaleError               1.01   0.00615
## systemModels.external_in.likelihood.rate                1.05    0.0148
## systemModels.external_in.prior.mean                     1.01     -4.17
## systemModels.external_in.hyper.time.scaleLevel          1.01    0.0438
## systemModels.external_in.hyper.time.scaleError          1.01    0.0369
## systemModels.deaths.likelihood.rate                     1.01   0.00662
## systemModels.deaths.prior.mean                          1.01     -5.01
## systemModels.deaths.hyper.time.scaleLevel               1.00   0.00375
## systemModels.deaths.hyper.time.scaleError               1.06   0.00973
## systemModels.external_out.likelihood.rate               1.08    0.0126
## systemModels.external_out.prior.mean                    1.02     -4.34
## systemModels.external_out.hyper.time.scaleLevel         1.00    0.0116
## systemModels.external_out.hyper.time.scaleError         1.01    0.0526
## dataModels.population_admin_nation.likelihood.rate      1.03      1.01
## dataModels.population_admin_nation.prior.mean           1.02    0.0127
## dataModels.population_admin_nation.prior.sd             1.05 0.0000877
## dataModels.arrivals_nation.likelihood.rate              1.05     0.901
## dataModels.arrivals_nation.prior.mean                   1.01   -0.0166
## dataModels.arrivals_nation.prior.sd                     1.06   0.00634
## dataModels.departures_nation.likelihood.rate            1.07     0.853
## dataModels.departures_nation.prior.mean                 1.04   -0.0461
## dataModels.departures_nation.prior.sd                   1.09   0.00314
##                                                          50%   97.5% length
## account.population                                   4155326 4622621     21
## account.births                                         58466   64258     20
## account.external_in                                    82564  116170     20
## account.deaths                                         28459   31545     20
## account.external_out                                   68594   90934     20
## systemModels.population.likelihood.count             4155561 4623520     21
## systemModels.population.prior.mean                      15.2    15.3     21
## systemModels.population.hyper.time.scaleLevel       0.000814  0.0047      1
## systemModels.population.hyper.time.scaleTrend         0.0031 0.00535      1
## systemModels.population.hyper.time.scaleError        0.00145  0.0261      1
## systemModels.births.likelihood.rate                   0.0144  0.0152     20
## systemModels.births.prior.mean                         -4.24   -4.19     20
## systemModels.births.hyper.time.scaleLevel             0.0272  0.0449      1
## systemModels.births.hyper.time.scaleError             0.0202  0.0421      1
## systemModels.external_in.likelihood.rate              0.0195  0.0255     20
## systemModels.external_in.prior.mean                    -3.95   -3.68     20
## systemModels.external_in.hyper.time.scaleLevel        0.0937   0.152      1
## systemModels.external_in.hyper.time.scaleError        0.0919   0.193      1
## systemModels.deaths.likelihood.rate                  0.00689 0.00732     20
## systemModels.deaths.prior.mean                         -4.98   -4.93     20
## systemModels.deaths.hyper.time.scaleLevel            0.00984  0.0174      1
```

```
## systemModels.deaths.hyper.time.scaleError             0.016  0.0323      1
## systemModels.external_out.likelihood.rate            0.0169  0.0208     20
## systemModels.external_out.prior.mean                  -4.08   -3.91     20
## systemModels.external_out.hyper.time.scaleLevel      0.0787   0.135      1
## systemModels.external_out.hyper.time.scaleError      0.0952   0.159      1
## dataModels.population_admin_nation.likelihood.rate     1.02    1.03     10
## dataModels.population_admin_nation.prior.mean         0.018  0.0253     10
## dataModels.population_admin_nation.prior.sd          0.0011 0.00454      1
## dataModels.arrivals_nation.likelihood.rate             1.02    1.15     20
## dataModels.arrivals_nation.prior.mean                0.0209  0.0488     20
## dataModels.arrivals_nation.prior.sd                   0.056  0.0869      1
## dataModels.departures_nation.likelihood.rate          0.992    1.11     20
## dataModels.departures_nation.prior.mean            -0.00848  0.0176     20
## dataModels.departures_nation.prior.sd                0.0485   0.105      1
## ------------------------------------------------------
```

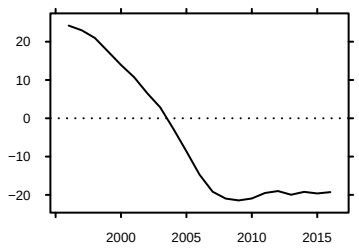
Results from the Alternative Model

```
#National population estimates
popn_sens <- dbind(Baseline = fetch(filename, where = c("account", "population")),
                           Sensitivity = fetch(filename_sens, where = c("account", "population")),
                           along = "variant")

dplot(~ time | variant,
      data = 1e-6 * popn_sens,
      col = "grey70",
      xlab = "",
      ylab = "",
      as.table = TRUE,
      scales = list(tck = 0.3),
      par.settings = list(fontsize = list(text = 6),
                           strip.background = list(col = "grey90")))
```



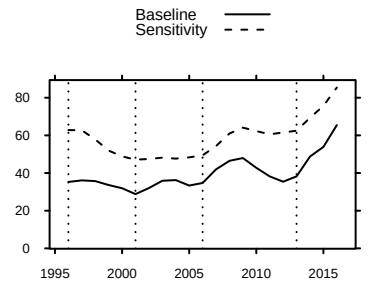

```
#Posterior median of alternative model minus posterior median of baseline model
popn_sens_median <- popn_sens %>%
    collapseIterations(FUN = median)

diff <- (subarray(popn_sens_median, variant == "Sensitivity")
    - subarray(popn_sens_median, variant == "Baseline")) %>%
    CountsOne(labels = names(.), name = "time", dimscale = "Points")

p <- dplot(~ time,
```

```
            data = 1e-3 * diff,
            xlab = "",
            ylab = "",
            as.table = TRUE,
            col = "black",
            scales = list(tck = 0.3),
            par.settings = list(fontsize = list(text = 6),
                                strip.background = list(col = "grey90")))
p <- p + latticeExtra::layer(panel.abline(h = 0,
                                          col = "black",
                                          lty = "dotted"))
plot(p)
```




```
#Widths of 95% credible intervals
popn_cred_int <- popn_sens %>%
    collapseIterations(prob = c(0.025, 0.975))

widths <- (slab(popn_cred_int, dimension = "quantile", element = 2) -
    slab(popn_cred_int, dimension = "quantile", element = 1)) %>%
    as.data.frame(direction = "long") %>%
    mutate(count = 1e-3 * count)

lty <- c("solid", "dashed")
p <- xyplot(count ~ time,
            data = widths,
            groups = variant,
            xlab = "",
            ylab = "",
            type = "l",
            as.table = TRUE,
            lty = lty,
            col = "black",
            ylim = c(-0.2, NA),
            scales = list(tck = 0.3),
            lattice.options = modifyList(lattice.options(),
                                         list(skip.boundary.labels = 0)),
            par.settings = list(fontsize = list(text = 6),
                                strip.background = list(col = "grey90")),
            key = list(text = list(levels(widths$variant)),
                       lines = list(col = "black", lty = lty),
                       space = "top"))
p <- p + latticeExtra::layer(panel.abline(v = c(1996, 2001, 2006, 2013),
                                          lty = "dotted",
```

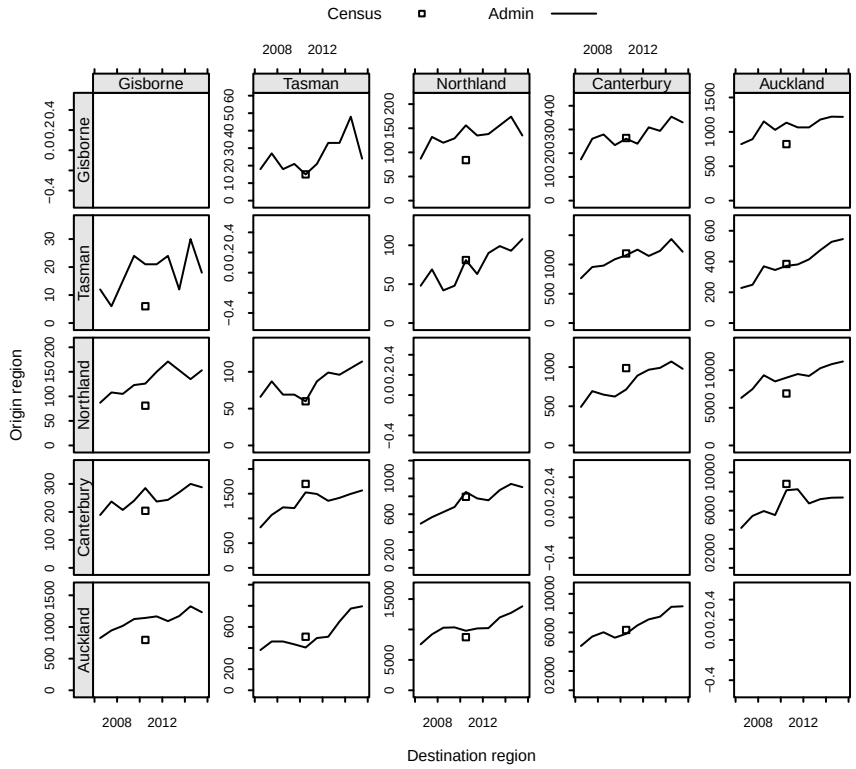
```
                                          col = "black",
                                          under = TRUE))
plot(p)
```



Input Data for the Regional Demographic Account

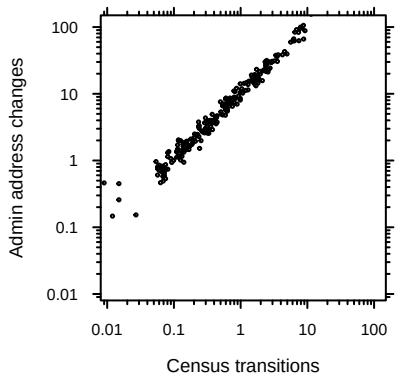
```
#Data on internal migration for five selected regions
prepanel <-  function(y) {
    if (all(is.na(y)))
        list(ylim = c(-0.5, 0.5))
    else
        list(ylim = c(0, 1.2 * max(y, na.rm = TRUE)))
}

address_admin_df <- address_admin %>%
    as.data.frame(direction = "long", midpoints = "time") %>%
    mutate(variant = "Admin")

address_census_df <- address_census %>%
    as.data.frame(direction = "long", midpoints = "time") %>%
    mutate(variant = "Census")

sample_regions <- c("Northland", "Auckland", "Gisborne", "Tasman", "Canterbury")

address_change <- bind_rows(address_admin_df, address_census_df) %>%
    mutate(count = ifelse(region_orig == region_dest, NA, count)) %>%
    filter((region_orig %in% sample_regions) & (region_dest %in% sample_regions)) %>%
    mutate(variant = factor(variant,
                            levels = c("Census", "Admin"),
                            labels = c("Census", "Admin")))


pch <- c(0L, NA_integer_)
lty <- c("blank", "solid")
p <- xyplot(count ~ time | region_dest + region_orig,
            data = address_change,
            groups = variant,
            as.table = TRUE,
            scales = list(tck = 0.3,
                          y = list(relation = "free")),
            prepanel = prepanel,
            par.settings = list(fontsize = list(text = 6, points = 4),
```

```
                                  strip.background = list(col = "grey90")),
            type = "b",
            pch = pch,
            lty = lty,
            between = list(y = 0.2),
            xlab = "Destination region",
            ylab = "Origin region",
            col = "black",
            key = list(text = list(levels(address_change$variant)),
                       lines = list(lty = lty,
                                    pch = pch,
                                    col = "black",
                                    type = "b",
                                    cex = 0.7),
                       divide = 1,
                       columns = 2))
p <- useOuterStrips(p)
plot(p)
```



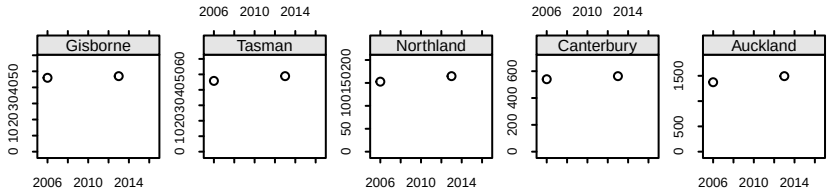

```
#Administrative address changes versus census 5-year transitions
address_admin_df <- address_admin %>%
    collapseDimension(margin = "region") %>%
    as.data.frame(direction = "long", responseName = "Admin") %>%
    mutate(Admin = Admin / 1000)

address_census_df <- address_census %>%
    collapseDimension(margin = "region", na.rm = TRUE) %>%
    as.data.frame(direction = "long", responseName = "Census") %>%
```

```
    mutate(Census = Census / 1000)

address_change <- address_admin_df %>%
    left_join(address_census_df, by = c("region_orig", "region_dest")) %>%
    filter(region_orig != region_dest)

p <- xyplot(Admin ~ Census,
            data = address_change,
            scales = list(tck = 0.5, log = TRUE),
            par.settings = list(fontsize = list(text = 7, points = 2),
                                strip.background = list(col = "grey90")),
            type = "p",
            xlab = "Census transitions",
            ylab = "Admin address changes",
            col = "black",
            aspect = 1,
            ylim = c(0.008, 150),
            xlim = c(0.008, 150),
            xscale.components = xscale.components.log10ticks,
            yscale.components = yscale.components.log10ticks)
plot(p)
```



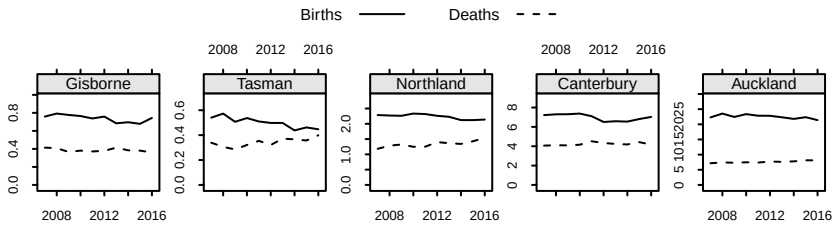

```
#Census-based population estimates
popn <- population_census %>%
    subarray(time >= 2006) %>%
    subarray(region %in% sample_regions) %>%
    as.data.frame(direction = "long") %>%
    mutate(count = 1e-3 * count)


xyplot(count ~ time | region,
       data = popn,
       as.table = TRUE,
       scales = list(tck = 0.3,
                     y = list(relation = "free")),
       xlim = c(2005, 2017),
       prepanel = prepanel,
       par.settings = list(fontsize = list(text = 6, points = 5),
                           strip.background = list(col = "grey90")),
```

```
       pch = 1,
       type = "p",
       layout = c(NA, 1L),
       xlab = "",
       ylab = "",
       col = "black")
```

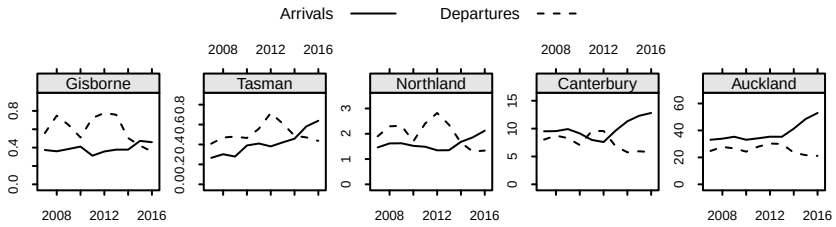



```
#Registered births and deaths
births_reg_df <- births_reg %>%
    subarray(time > 2006) %>%
    subarray(region %in% sample_regions) %>%
    collapseDimension(margin = c("region", "time")) %>%
    as.data.frame(direction = "long") %>%
    mutate(dataset = "Births")

deaths_reg_df <- deaths_reg %>%
    subarray(time > 2006) %>%
    subarray(region %in% sample_regions) %>%
    collapseDimension(margin = c("region", "time")) %>%
    as.data.frame(direction = "long") %>%
    mutate(dataset = "Deaths")

combined <- bind_rows(births_reg_df, deaths_reg_df) %>%
    mutate(dataset = factor(dataset, levels = c("Births", "Deaths"))) %>%
    mutate(time = as.integer(levels(time))[time]) %>%
    mutate(count = 1e-3 * count)


lty <- c("solid", "dashed")
xyplot(count ~ time | region,
       data = combined,
       groups = dataset,
       as.table = TRUE,
       scales = list(tck = 0.3,
                     y = list(relation = "free")),
       par.settings = list(fontsize = list(text = 6),
                           strip.background = list(col = "grey90")),
       type = "l",
       lty = lty,
       layout = c(NA, 1L),
       xlab = "",
       ylab = "",
       prepanel = prepanel,
       col = "black",
       key = list(text = list(levels(combined$dataset)),
```

```
                  lines = list(lty = lty,
                               col = "black"),
                  columns = 2))
```




```
#International arrivals and departures
arrivals_df <- arrivals %>%
    subarray(time > 2006) %>%
    subarray(region %in% sample_regions) %>%
    as.data.frame(direction = "long") %>%
    mutate(dataset = "Arrivals")

departures_df <- departures %>%
    subarray(time > 2006) %>%
    subarray(region %in% sample_regions) %>%
    as.data.frame(direction = "long") %>%
    mutate(dataset = "Departures")

combined <- bind_rows(arrivals_df, departures_df) %>%
    mutate(dataset = factor(dataset, levels = c("Arrivals", "Departures"))) %>%
    mutate(time = as.integer(levels(time))[time]) %>%
    mutate(count = 1e-3 * count)

lty <- c("solid", "dashed")
xyplot(count ~ time | region,
       data = combined,
       groups = dataset,
       as.table = TRUE,
       scales = list(tck = 0.3,
                     y = list(relation = "free")),
       par.settings = list(fontsize = list(text = 6),
                           strip.background = list(col = "grey90")),
       type = "l",
       lty = lty,
       layout = c(NA, 1L),
       xlab = "",
       ylab = "",
       prepanel = prepanel,
       col = "black",
       key = list(text = list(levels(combined$dataset)),
                  lines = list(lty = lty,
                               col = "black"),
                  columns = 2))
```



Fit model for Regional Demographic Account

```
filename_region <- "new_zealand_region.est"
if(!file.exists(filename_region)) {
    #Get datasets
    arrivals <- arrivals %>% subarray(region != "Not Stated")
    departures <- departures %>% subarray(region != "Not Stated")
    datasets <-
        list(population_census = subarray(population_census, time >= 2006),
             population_admin = subarray(population_admin, time >= 2006, drop = FALSE),
             births_reg = subarray(births_reg, time >= 2006),
             deaths_reg = subarray(deaths_reg, time >= 2006),
             arrivals = subarray(arrivals, time >= 2006),
             departures = subarray(departures, time >= 2006),
             address_census = address_census,
             address_admin = subarray(address_admin, time >= 2006))

    #Initial account
    growth <- growth(population_census, type = "linear", within = "region")
    population <- population_census %>%
        subarray(time == "2013", drop = FALSE) %>%
        extrapolate(labels = 2014:2016, growth = growth, type = "linear") %>%
        extrapolate(labels = 2006:2012, growth = growth, type = "linear") %>%
        toInteger(force = TRUE)

    internal <- (0.75 * address_admin) %>%
        toInteger(force = TRUE)

    account <- Movements(population = population,
                         births = births_reg,
                         internal = internal,
                         entries = list(external_in = arrivals),
                         exits = list(deaths = deaths_reg,
                                      external_out = departures)) %>%
        makeConsistent()


    #Set up system models
    log_popn <- population_census %>%
        subarray(time == "2013", drop = FALSE) %>%
        collapseDimension(margin = "region") %>%
        as.data.frame(direction = "long") %>%
        mutate(log_popn = log(count)) %>%
        select(region, log_popn)
```

```
systemModels <- list(
    Model(population ~ Poisson(mean ~ region * time, useExpose = FALSE),
          time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                     trend = Trend(scale = HalfT(scale = 0.05)),
                     damp = NULL,
                     error = Error(scale = HalfT(scale = 0.05))),
          region:time ~ DLM(level = Level(scale = HalfT(scale = 0.025)),
                            trend = Trend(scale = HalfT(scale = 0.025)),
                            damp = NULL,
                            error = Error(scale = HalfT(scale = 0.025))),
          jump = 0.002),
    Model(births ~ Poisson(mean ~ region + time),
          time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                     trend = NULL,
                     damp = NULL,
                     error = Error(scale = HalfT(scale = 0.05))),
          jump = 0.03),
    Model(deaths ~ Poisson(mean ~ region + time),
          time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                     trend = Trend(scale = HalfT(scale = 0.05)),
                     damp = NULL,
                     error = Error(scale = HalfT(scale = 0.05))),
          jump = 0.04),
    Model(external_in ~ Poisson(mean ~ region + time),
          time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                     trend = NULL,
                     damp = NULL,
                     error = Error(scale = HalfT(scale = 0.05))),
          jump = 0.03),
    Model(external_out ~ Poisson(mean ~ region + time),
           time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                      trend = NULL,
                      damp = NULL,
                      error = Error(scale = HalfT(scale = 0.05))),
           jump = 0.02),
    Model(internal ~ Poisson(mean ~ region_orig * region_dest + time,
                             structuralZeros = "diag"),
          region_orig ~ Exch(covariates = Covariates(mean ~ log_popn, data = log_popn)),
          region_dest ~ Exch(covariates = Covariates(mean ~ log_popn, data = log_popn)),
          region_orig:region_dest ~ Exch(),
          time ~ DLM(level = Level(scale = HalfT(scale = 0.05)),
                     trend = NULL,
                     damp = NULL,
                     error = Error(scale = HalfT(scale = 0.05))),
          jump = 0.04))

#Set up data models
sd <- 0.0025 * as(population_census, "Values")
mean <- sd / sd
dataModels <- list(
    Model(population_census ~ NormalFixed(mean = mean, sd = sd),
          series = "population"),
    Model(population_admin ~ Poisson(mean ~ 1),
```

```
            `(Intercept)` ~ ExchFixed(sd = 0.025),
            series = "population",
            priorSD = HalfT(scale = 0.025),
            jump = 0.0014),
      Model(births_reg ~ PoissonBinomial(prob = 0.98),
            series = "births"),
      Model(deaths_reg ~ PoissonBinomial(prob = 0.99),
            series = "deaths"),
      Model(arrivals ~ Poisson(mean ~ 1),
            `(Intercept)` ~ ExchFixed(sd = 0.025),
            lower = 0.8,
            upper = 1.2,
            priorSD = HalfT(scale = 0.025),
            series = "external_in",
            jump = 0.04),
      Model(departures ~ Poisson(mean ~ 1),
            `(Intercept)` ~ ExchFixed(sd = 0.025),
            lower = 0.8,
            upper = 1.2,
            priorSD = HalfT(scale = 0.025),
            series = "external_out",
            jump = 0.04),
      Model(address_census ~ Poisson(mean ~ 1,
                                     structuralZeros = "diag"),
            priorSD = HalfT(scale = 0.1),
            series = "internal",
            lower = 0.1,
            upper = 0.3,
            jump = 0.028),
      Model(address_admin ~ Poisson(mean ~ 1,
                                    structuralZeros = "diag"),
            priorSD = HalfT(scale = 0.1),
            series = "internal",
            lower = 0.7,
            upper = 2,
            jump = 0.055))


#Estimate account.
n_burnin <- 200000
n_sim <- 200000
n_chain <- 4
n_thin <- 500

set.seed(0)

estimateAccount(account = account,
              systemModels = systemModels,
              datasets = datasets,
              dataModels = dataModels,
              filename = filename_region,
              nBurnin = n_burnin,
              nSim = n_sim,
```

```
                nChain = n_chain,
                nThin = n_thin)
}
fetchSummary(filename_region)
```

```
## ------------------------------------------------
## Account:
## *population*
## dimensions: region, time
## n cells: 176
## *births*
## dimensions: region, time
## n cells: 160
## *internal*
## dimensions: region_orig, region_dest, time
## n cells: 2560
## *external_in*
## dimensions: region, time
## n cells: 160
## *deaths*
## dimensions: region, time
## n cells: 160
## *external_out*
## dimensions: region, time
## n cells: 160
## ------------------------------------------------
## System models:
## *population*
## population ~ Poisson(mean ~ region * time, useExpose = FALSE),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = Trend(scale = HalfT(scale = 0.05)),
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## region:time ~ DLM(level = Level(scale = HalfT(scale = 0.025)),
##     trend = Trend(scale = HalfT(scale = 0.025)), damp = NULL,
##     error = Error(scale = HalfT(scale = 0.025))),
## 0.002
## dimensions: region, time
## *births*
## births ~ Poisson(mean ~ region + time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.03
## dimensions: region, time
## *internal*
## internal ~ Poisson(mean ~ region_orig * region_dest + time, structuralZeros = "diag"),
## region_orig ~ Exch(covariates = Covariates(mean ~ log_popn, data = log_popn)),
## region_dest ~ Exch(covariates = Covariates(mean ~ log_popn, data = log_popn)),
## region_orig:region_dest ~ Exch(),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.04
## dimensions: region_orig, region_dest, time
## *external_in*
## external_in ~ Poisson(mean ~ region + time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = NULL,
```

```
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.03
## dimensions: region, time
## *deaths*
## deaths ~ Poisson(mean ~ region + time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = Trend(scale = HalfT(scale = 0.05)),
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.04
## dimensions: region, time
## *external_out*
## external_out ~ Poisson(mean ~ region + time),
## time ~ DLM(level = Level(scale = HalfT(scale = 0.05)), trend = NULL,
##     damp = NULL, error = Error(scale = HalfT(scale = 0.05))),
## 0.02
## dimensions: region, time
## ------------------------------------------------
## Datasets:
## *population_census*
## Object of class "Counts"
## dimensions: region, time
## n cells: 32,  n missing: 0, integers: TRUE, n zeros: 0, median: 155330
## *population_admin*
## Object of class "Counts"
## dimensions: time
## n cells: 10,  n missing: 0, integers: TRUE, n zeros: 0, median: 4451648
## *births_reg*
## Object of class "Counts"
## dimensions: region, time
## n cells: 160,  n missing: 0, integers: TRUE, n zeros: 0, median: 2220
## *deaths_reg*
## Object of class "Counts"
## dimensions: region, time
## n cells: 160,  n missing: 0, integers: TRUE, n zeros: 0, median: 1278
## *arrivals*
## Object of class "Counts"
## dimensions: region, time
## n cells: 160,  n missing: 0, integers: TRUE, n zeros: 0, median: 1620.5
## *departures*
## Object of class "Counts"
## dimensions: region, time
## n cells: 160,  n missing: 0, integers: TRUE, n zeros: 0, median: 1948
## *address_census*
## Object of class "Counts"
## dimensions: region_orig, region_dest, time
## n cells: 256,  n missing: 0, integers: TRUE, n zeros: 17, median: 381
## *address_admin*
## Object of class "Counts"
## dimensions: region_orig, region_dest, time
## n cells: 2560,  n missing: 0, integers: TRUE, n zeros: 160, median: 399
## ------------------------------------------------
## Data models:
## *population_census*
## population_census ~ NormalFixed(mean = mean, sd = sd),
## "population"
```

```
## dimensions: region, time
## *population_admin*
## population_admin ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.025),
## HalfT(scale = 0.025),
## 0.0014,
## "population"
## dimensions: time
## *births_reg*
## births_reg ~ PoissonBinomial(prob = 0.98),
## "births"
## dimensions: region, time
## *deaths_reg*
## deaths_reg ~ PoissonBinomial(prob = 0.99),
## "deaths"
## dimensions: region, time
## *arrivals*
## arrivals ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.025),
## 0.8,
## 1.2,
## HalfT(scale = 0.025),
## 0.04,
## "external_in"
## dimensions: region, time
## *departures*
## departures ~ Poisson(mean ~ 1),
## `(Intercept)` ~ ExchFixed(sd = 0.025),
## 0.8,
## 1.2,
## HalfT(scale = 0.025),
## 0.04,
## "external_out"
## dimensions: region, time
## *address_census*
## address_census ~ Poisson(mean ~ 1, structuralZeros = "diag"),
## 0.1,
## 0.3,
## HalfT(scale = 0.1),
## 0.028,
## "internal"
## dimensions: region_orig, region_dest, time
## *address_admin*
## address_admin ~ Poisson(mean ~ 1, structuralZeros = "diag"),
## 0.7,
## 2,
## HalfT(scale = 0.1),
## 0.055,
## "internal"
## dimensions: region_orig, region_dest, time
## ------------------------------------------------
## MCMC statistics:
## nBurnin: 200000,  nSim: 200000,  nChain: 4,  nThin: 500,  nIteration: 1600
##
```

```
## Metropolis-Hastings updates:
##                                                jump acceptance autocorr
## systemModels.population.likelihood.count      0.002      0.250    0.599
## systemModels.births.likelihood.rate           0.030      0.356    0.042
## systemModels.internal.likelihood.rate         0.040      0.406    0.159
## systemModels.external_in.likelihood.rate      0.030      0.433    0.141
## systemModels.deaths.likelihood.rate           0.040      0.360    0.042
## systemModels.external_out.likelihood.rate     0.020      0.393    0.312
## dataModels.population_admin.likelihood.rate   0.001      0.363    0.400
## dataModels.arrivals.likelihood.rate           0.040      0.399    0.102
## dataModels.departures.likelihood.rate         0.040      0.431    0.173
## dataModels.address_census.likelihood.rate     0.028      0.685    0.122
## dataModels.address_admin.likelihood.rate      0.055      0.397    0.134
##
## parameters:
##                                                                  Rhat
## account.population                                               1.04
## account.births                                                   1.01
## account.internal                                                 1.03
## account.external_in                                              1.02
## account.deaths                                                   1.00
## account.external_out                                             1.03
## systemModels.population.likelihood.count                       . 1.13
## systemModels.population.prior.mean                               1.08
## systemModels.population.hyper.region.scaleError                  1.03
## systemModels.population.hyper.time.scaleLevel                    1.00
## systemModels.population.hyper.time.scaleTrend                    1.01
## systemModels.population.hyper.time.scaleError                    1.00
## systemModels.population.hyper.region:time.scaleLevel             1.01
## systemModels.population.hyper.region:time.scaleTrend             1.02
## systemModels.population.hyper.region:time.scaleError             1.00
## systemModels.births.likelihood.rate                              1.00
## systemModels.births.prior.mean                                   1.01
## systemModels.births.prior.sd                                     1.01
## systemModels.births.hyper.region.scaleError                      1.00
## systemModels.births.hyper.time.scaleLevel                        1.00
## systemModels.births.hyper.time.scaleError                        1.01
## systemModels.internal.likelihood.rate                            1.03
## systemModels.internal.prior.mean                                 1.03
## systemModels.internal.prior.sd                                   1.06
## systemModels.internal.hyper.region_orig.coef                     1.07
## systemModels.internal.hyper.region_orig.scaleError             . 1.16
## systemModels.internal.hyper.region_dest.coef                     1.06
## systemModels.internal.hyper.region_dest.scaleError             . 1.17
## systemModels.internal.hyper.time.scaleLevel                      1.00
## systemModels.internal.hyper.time.scaleError                      1.00
## systemModels.internal.hyper.region_orig:region_dest.scaleError   1.01
## systemModels.external_in.likelihood.rate                         1.05
## systemModels.external_in.prior.mean                              1.06
## systemModels.external_in.prior.sd                                1.01
## systemModels.external_in.hyper.region.scaleError                 1.00
## systemModels.external_in.hyper.time.scaleLevel                   1.00
## systemModels.external_in.hyper.time.scaleError                   1.01
## systemModels.deaths.likelihood.rate                              1.01
```

```
## systemModels.deaths.prior.mean                                    1.00
## systemModels.deaths.prior.sd                                      1.00
## systemModels.deaths.hyper.region.scaleError                       1.00
## systemModels.deaths.hyper.time.scaleLevel                         1.00
## systemModels.deaths.hyper.time.scaleTrend                         1.01
## systemModels.deaths.hyper.time.scaleError                         1.00
## systemModels.external_out.likelihood.rate                         1.03
## systemModels.external_out.prior.mean                              1.02
## systemModels.external_out.prior.sd                                1.01
## systemModels.external_out.hyper.region.scaleError                 1.00
## systemModels.external_out.hyper.time.scaleLevel                   1.00
## systemModels.external_out.hyper.time.scaleError                   1.00
## dataModels.population_admin.likelihood.rate                       1.01
## dataModels.population_admin.prior.mean                            1.01
## dataModels.population_admin.prior.sd                              1.00
## dataModels.arrivals.likelihood.rate                               1.03
## dataModels.arrivals.prior.mean                                    1.01
## dataModels.arrivals.prior.sd                                      1.02
## dataModels.departures.likelihood.rate                             1.02
## dataModels.departures.prior.mean                                  1.01
## dataModels.departures.prior.sd                                    1.01
## dataModels.address_census.likelihood.rate                         1.03
## dataModels.address_census.prior.mean                              1.06
## dataModels.address_census.prior.sd                                1.00
## dataModels.address_admin.likelihood.rate                          1.02
## dataModels.address_admin.prior.mean                             . 1.10
## dataModels.address_admin.prior.sd                                 1.03
##                                                                    2.5%
## account.population                                                32782
## account.births                                                      407
## account.internal                                                      0
## account.external_in                                                 295
## account.deaths                                                      269
## account.external_out                                                296
## systemModels.population.likelihood.count                          32752
## systemModels.population.prior.mean                                 10.4
## systemModels.population.hyper.region.scaleError                   0.842
## systemModels.population.hyper.time.scaleLevel                  0.000173
## systemModels.population.hyper.time.scaleTrend                   0.00112
## systemModels.population.hyper.time.scaleError                 0.0000653
## systemModels.population.hyper.region:time.scaleLevel          0.0000217
## systemModels.population.hyper.region:time.scaleTrend          0.0000532
## systemModels.population.hyper.region:time.scaleError           0.000252
## systemModels.births.likelihood.rate                              0.0101
## systemModels.births.prior.mean                                    -4.59
## systemModels.births.prior.sd                                     0.0136
## systemModels.births.hyper.region.scaleError                      0.0841
## systemModels.births.hyper.time.scaleLevel                        0.0181
## systemModels.births.hyper.time.scaleError                       0.00192
## systemModels.internal.likelihood.rate                          0.000371
## systemModels.internal.prior.mean                                  -7.91
## systemModels.internal.prior.sd                                  0.00758
## systemModels.internal.hyper.region_orig.coef                     -0.592
## systemModels.internal.hyper.region_orig.scaleError              0.00919
```

```
## systemModels.internal.hyper.region_dest.coef                        1.46
## systemModels.internal.hyper.region_dest.scaleError                 0.015
## systemModels.internal.hyper.time.scaleLevel                       0.0145
## systemModels.internal.hyper.time.scaleError                       0.0025
## systemModels.internal.hyper.region_orig:region_dest.scaleError     0.769
## systemModels.external_in.likelihood.rate                         0.00766
## systemModels.external_in.prior.mean                                -4.85
## systemModels.external_in.prior.sd                                 0.0189
## systemModels.external_in.hyper.region.scaleError                   0.252
## systemModels.external_in.hyper.time.scaleLevel                    0.0443
## systemModels.external_in.hyper.time.scaleError                     0.004
## systemModels.deaths.likelihood.rate                              0.00516
## systemModels.deaths.prior.mean                                     -5.27
## systemModels.deaths.prior.sd                                      0.0137
## systemModels.deaths.hyper.region.scaleError                       0.0989
## systemModels.deaths.hyper.time.scaleLevel                       0.000139
## systemModels.deaths.hyper.time.scaleTrend                        0.00054
## systemModels.deaths.hyper.time.scaleError                        0.00122
## systemModels.external_out.likelihood.rate                        0.00739
## systemModels.external_out.prior.mean                               -4.91
## systemModels.external_out.prior.sd                               0.00342
## systemModels.external_out.hyper.region.scaleError                  0.139
## systemModels.external_out.hyper.time.scaleLevel                   0.0703
## systemModels.external_out.hyper.time.scaleError                  0.00731
## dataModels.population_admin.likelihood.rate                         1.01
## dataModels.population_admin.prior.mean                           0.0105
## dataModels.population_admin.prior.sd                             0.00146
## dataModels.arrivals.likelihood.rate                                0.834
## dataModels.arrivals.prior.mean                                   -0.0619
## dataModels.arrivals.prior.sd                                      0.0489
## dataModels.departures.likelihood.rate                              0.857
## dataModels.departures.prior.mean                                 -0.0138
## dataModels.departures.prior.sd                                     0.067
## dataModels.address_census.likelihood.rate                          0.143
## dataModels.address_census.prior.mean                               -1.63
## dataModels.address_census.prior.sd                                 0.157
## dataModels.address_admin.likelihood.rate                           0.962
## dataModels.address_admin.prior.mean                               0.0468
## dataModels.address_admin.prior.sd                                 0.0519
##                                                                      50%
## account.population                                                156997
## account.births                                                      2221
## account.internal                                                     373
## account.external_in                                                 1696
## account.deaths                                                      1280
## account.external_out                                                1978
## systemModels.population.likelihood.count                          156955
## systemModels.population.prior.mean                                    12
## systemModels.population.hyper.region.scaleError                     1.04
## systemModels.population.hyper.time.scaleLevel                     0.00166
## systemModels.population.hyper.time.scaleTrend                     0.00317
## systemModels.population.hyper.time.scaleError                    0.000873
## systemModels.population.hyper.region:time.scaleLevel             0.000411
## systemModels.population.hyper.region:time.scaleTrend             0.000683
```

```
## systemModels.population.hyper.region:time.scaleError           0.000497
## systemModels.births.likelihood.rate                              0.0133
## systemModels.births.prior.mean                                   -4.32
## systemModels.births.prior.sd                                     0.0165
## systemModels.births.hyper.region.scaleError                       0.119
## systemModels.births.hyper.time.scaleLevel                        0.0279
## systemModels.births.hyper.time.scaleError                        0.0061
## systemModels.internal.likelihood.rate                           0.00304
## systemModels.internal.prior.mean                                   -5.8
## systemModels.internal.prior.sd                                    0.029
## systemModels.internal.hyper.region_orig.coef                      -0.35
## systemModels.internal.hyper.region_orig.scaleError               0.0763
## systemModels.internal.hyper.region_dest.coef                       1.79
## systemModels.internal.hyper.region_dest.scaleError                0.097
## systemModels.internal.hyper.time.scaleLevel                      0.0563
## systemModels.internal.hyper.time.scaleError                      0.0365
## systemModels.internal.hyper.region_orig:region_dest.scaleError    0.815
## systemModels.external_in.likelihood.rate                         0.0126
## systemModels.external_in.prior.mean                               -4.37
## systemModels.external_in.prior.sd                                0.0497
## systemModels.external_in.hyper.region.scaleError                  0.325
## systemModels.external_in.hyper.time.scaleLevel                   0.0678
## systemModels.external_in.hyper.time.scaleError                   0.0224
## systemModels.deaths.likelihood.rate                              0.0081
## systemModels.deaths.prior.mean                                    -4.81
## systemModels.deaths.prior.sd                                     0.0209
## systemModels.deaths.hyper.region.scaleError                       0.141
## systemModels.deaths.hyper.time.scaleLevel                       0.00916
## systemModels.deaths.hyper.time.scaleTrend                       0.00641
## systemModels.deaths.hyper.time.scaleError                       0.00796
## systemModels.external_out.likelihood.rate                        0.0127
## systemModels.external_out.prior.mean                              -4.37
## systemModels.external_out.prior.sd                               0.0192
## systemModels.external_out.hyper.region.scaleError                 0.188
## systemModels.external_out.hyper.time.scaleLevel                   0.117
## systemModels.external_out.hyper.time.scaleError                  0.0427
## dataModels.population_admin.likelihood.rate                        1.01
## dataModels.population_admin.prior.mean                           0.0122
## dataModels.population_admin.prior.sd                             0.0018
## dataModels.arrivals.likelihood.rate                               0.954
## dataModels.arrivals.prior.mean                                  -0.0469
## dataModels.arrivals.prior.sd                                      0.07
## dataModels.departures.likelihood.rate                             1.01
## dataModels.departures.prior.mean                                0.00435
## dataModels.departures.prior.sd                                   0.0761
## dataModels.address_census.likelihood.rate                         0.207
## dataModels.address_census.prior.mean                             -1.58
## dataModels.address_census.prior.sd                                0.17
## dataModels.address_admin.likelihood.rate                          1.08
## dataModels.address_admin.prior.mean                              0.0761
## dataModels.address_admin.prior.sd                                0.0574
##                                                                   97.5%
## account.population                                             1483193
## account.births                                                   22503
```

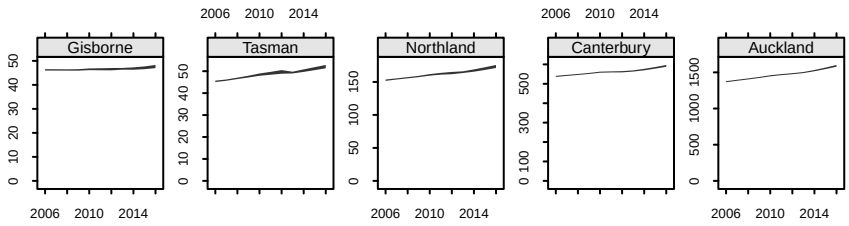
```
## account.internal                                                 8419
## account.external_in                                             36905
## account.deaths                                                   7649
## account.external_out                                            28765
## systemModels.population.likelihood.count                      1482879
## systemModels.population.prior.mean                              14.2
## systemModels.population.hyper.region.scaleError                  1.26
## systemModels.population.hyper.time.scaleLevel                 0.00446
## systemModels.population.hyper.time.scaleTrend                 0.00979
## systemModels.population.hyper.time.scaleError                 0.00254
## systemModels.population.hyper.region:time.scaleLevel          0.00118
## systemModels.population.hyper.region:time.scaleTrend          0.00131
## systemModels.population.hyper.region:time.scaleError          0.00108
## systemModels.births.likelihood.rate                            0.0166
## systemModels.births.prior.mean                                   -4.1
## systemModels.births.prior.sd                                   0.0214
## systemModels.births.hyper.region.scaleError                     0.177
## systemModels.births.hyper.time.scaleLevel                      0.0545
## systemModels.births.hyper.time.scaleError                      0.0285
## systemModels.internal.likelihood.rate                          0.0329
## systemModels.internal.prior.mean                                -3.41
## systemModels.internal.prior.sd                                 0.0434
## systemModels.internal.hyper.region_orig.coef                   -0.153
## systemModels.internal.hyper.region_orig.scaleError              0.206
## systemModels.internal.hyper.region_dest.coef                      2.1
## systemModels.internal.hyper.region_dest.scaleError              0.314
## systemModels.internal.hyper.time.scaleLevel                    0.0924
## systemModels.internal.hyper.time.scaleError                    0.0804
## systemModels.internal.hyper.region_orig:region_dest.scaleError   0.887
## systemModels.external_in.likelihood.rate                       0.0258
## systemModels.external_in.prior.mean                             -3.64
## systemModels.external_in.prior.sd                              0.0723
## systemModels.external_in.hyper.region.scaleError                0.602
## systemModels.external_in.hyper.time.scaleLevel                  0.113
## systemModels.external_in.hyper.time.scaleError                 0.0677
## systemModels.deaths.likelihood.rate                           0.00879
## systemModels.deaths.prior.mean                                  -4.74
## systemModels.deaths.prior.sd                                   0.0254
## systemModels.deaths.hyper.region.scaleError                     0.197
## systemModels.deaths.hyper.time.scaleLevel                      0.0273
## systemModels.deaths.hyper.time.scaleTrend                      0.0196
## systemModels.deaths.hyper.time.scaleError                      0.0272
## systemModels.external_out.likelihood.rate                      0.0202
## systemModels.external_out.prior.mean                            -3.91
## systemModels.external_out.prior.sd                             0.0407
## systemModels.external_out.hyper.region.scaleError               0.257
## systemModels.external_out.hyper.time.scaleLevel                  0.18
## systemModels.external_out.hyper.time.scaleError                 0.116
## dataModels.population_admin.likelihood.rate                      1.02
## dataModels.population_admin.prior.mean                          0.014
## dataModels.population_admin.prior.sd                          0.00236
## dataModels.arrivals.likelihood.rate                               1.1
## dataModels.arrivals.prior.mean                                -0.0157
## dataModels.arrivals.prior.sd                                   0.0854
```

```
## dataModels.departures.likelihood.rate                              1.17
## dataModels.departures.prior.mean                                 0.0247
## dataModels.departures.prior.sd                                   0.0858
## dataModels.address_census.likelihood.rate                         0.281
## dataModels.address_census.prior.mean                              -1.54
## dataModels.address_census.prior.sd                                0.188
## dataModels.address_admin.likelihood.rate                           1.22
## dataModels.address_admin.prior.mean                                0.12
## dataModels.address_admin.prior.sd                                0.0642
##                                                                  length
## account.population                                                  176
## account.births                                                      160
## account.internal                                                   2560
## account.external_in                                                 160
## account.deaths                                                      160
## account.external_out                                                160
## systemModels.population.likelihood.count                            176
## systemModels.population.prior.mean                                  176
## systemModels.population.hyper.region.scaleError                       1
## systemModels.population.hyper.time.scaleLevel                         1
## systemModels.population.hyper.time.scaleTrend                         1
## systemModels.population.hyper.time.scaleError                         1
## systemModels.population.hyper.region:time.scaleLevel                  1
## systemModels.population.hyper.region:time.scaleTrend                  1
## systemModels.population.hyper.region:time.scaleError                  1
## systemModels.births.likelihood.rate                                 160
## systemModels.births.prior.mean                                      160
## systemModels.births.prior.sd                                          1
## systemModels.births.hyper.region.scaleError                           1
## systemModels.births.hyper.time.scaleLevel                             1
## systemModels.births.hyper.time.scaleError                             1
## systemModels.internal.likelihood.rate                              2560
## systemModels.internal.prior.mean                                   2560
## systemModels.internal.prior.sd                                        1
## systemModels.internal.hyper.region_orig.coef                          1
## systemModels.internal.hyper.region_orig.scaleError                    1
## systemModels.internal.hyper.region_dest.coef                          1
## systemModels.internal.hyper.region_dest.scaleError                    1
## systemModels.internal.hyper.time.scaleLevel                           1
## systemModels.internal.hyper.time.scaleError                           1
## systemModels.internal.hyper.region_orig:region_dest.scaleError        1
## systemModels.external_in.likelihood.rate                            160
## systemModels.external_in.prior.mean                                 160
## systemModels.external_in.prior.sd                                     1
## systemModels.external_in.hyper.region.scaleError                      1
## systemModels.external_in.hyper.time.scaleLevel                        1
## systemModels.external_in.hyper.time.scaleError                        1
## systemModels.deaths.likelihood.rate                                 160
## systemModels.deaths.prior.mean                                      160
## systemModels.deaths.prior.sd                                          1
## systemModels.deaths.hyper.region.scaleError                           1
## systemModels.deaths.hyper.time.scaleLevel                             1
## systemModels.deaths.hyper.time.scaleTrend                             1
## systemModels.deaths.hyper.time.scaleError                             1
```

```
## systemModels.external_out.likelihood.rate                    160
## systemModels.external_out.prior.mean                         160
## systemModels.external_out.prior.sd                             1
## systemModels.external_out.hyper.region.scaleError              1
## systemModels.external_out.hyper.time.scaleLevel                1
## systemModels.external_out.hyper.time.scaleError                1
## dataModels.population_admin.likelihood.rate                   10
## dataModels.population_admin.prior.mean                        10
## dataModels.population_admin.prior.sd                           1
## dataModels.arrivals.likelihood.rate                          160
## dataModels.arrivals.prior.mean                               160
## dataModels.arrivals.prior.sd                                   1
## dataModels.departures.likelihood.rate                        160
## dataModels.departures.prior.mean                             160
## dataModels.departures.prior.sd                                 1
## dataModels.address_census.likelihood.rate                    256
## dataModels.address_census.prior.mean                         256
## dataModels.address_census.prior.sd                             1
## dataModels.address_admin.likelihood.rate                    2560
## dataModels.address_admin.prior.mean                         2560
## dataModels.address_admin.prior.sd                              1
## ------------------------------------------------------
```

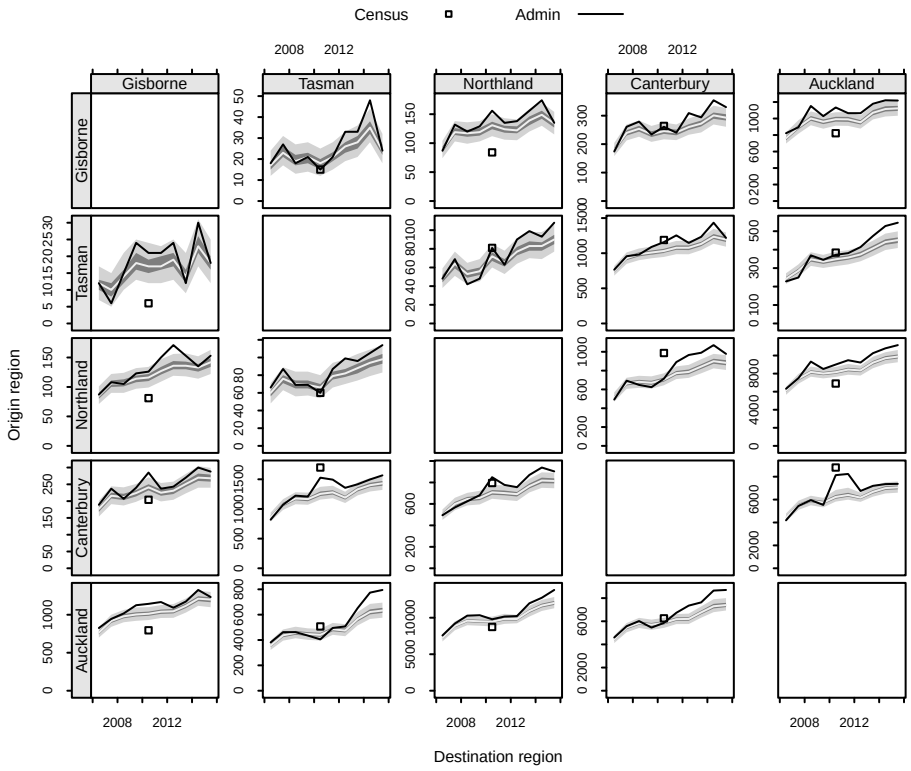
Results for the Regional Demographic Account

```
#Estimates of population (thousands) in five selected regions.
population <- fetch(filename_region, where = c("account", "population")) %>%
    subarray(region %in% sample_regions) %>%
    subarray(time != "2017")

dplot(~ time | region,
      data = 1e-3 * population,
      col = "grey20",
      xlab = "",
      ylab = "",
      prob = c(0.025, 0.975),
      as.table = TRUE,
      scales = list(tck = 0.3,
                    y = list(relation = "free")),
      ylim = c(0, NA),
      layout = c(NA, 1),
      par.settings = list(fontsize = list(text = 6),
                          strip.background = list(col = "grey90")))
```



```
#Estimates of internal migration between five selected regions.
internal <- fetch(filename_region, where = c("account", "internal")) %>%
    subarray((region_orig %in% sample_regions) & (region_dest %in% sample_regions))

admin <- fetch(filename_region, where = c("datasets", "address_admin")) %>%
    subarray((region_orig %in% sample_regions) & (region_dest %in% sample_regions)) %>%
    as.data.frame(direction = "long", midpoints = "time") %>%
    mutate(variant = "Admin")

census <- fetch(filename_region, where = c("datasets", "address_census")) %>%
    subarray((region_orig %in% sample_regions) & (region_dest %in% sample_regions)) %>%
    as.data.frame(direction = "long") %>%
    mutate(variant = "Census") %>%
    mutate(time=2010.5)

address_change <- bind_rows(admin, census) %>%
    mutate(count = ifelse(region_orig == region_dest, NA, count)) %>%
    mutate(variant = factor(variant,
                            levels = c("Census", "Admin"),
                            labels = c("Census", "Admin")))



pch <- c(0L, NA_integer_)
lty <- c("blank", "solid")

p <- xyplot(count ~ time | region_dest + region_orig,
            data = address_change,
            groups = variant,
            par.settings = list(fontsize = list(text = 6, points = 4),
                                strip.background = list(col = "grey90")),
            scales = list(tck = 0.3,
                          y = "free"),
            ylim = c(0, NA),
            as.table = TRUE,
            type = "b",
            pch = pch,
            lty = lty,
            between = list(y = 0.2),
            xlab = "Destination region",
            ylab = "Origin region",
            col = "black",
            key = list(text = list(levels(address_change$variant)),
                       lines = list(lty = lty,
                                    pch = pch,
                                    col = "black",
                                    type = "b",
                                    cex = 0.7),
                       divide = 1,
                       columns = 2))
p <- p + as.layer(dplot( ~ time | region_dest * region_orig,
                         data = internal,
                         col = "grey",
```

```
                        xlab = "",
                        ylab = "",
                        as.table = TRUE,
                        midpoints = "time",
                        na.rm = TRUE,
                        par.settings = list(fontsize = list(text = 6),
                                            strip.background = list(col = "grey90"))),
                    under = TRUE)
p <- useOuterStrips(p)
plot(p)
```




```
#Coverage ratios for the address change data.
admin <- fetchCoverage(filename_region, dataset = "address_admin") %>%
    subarray(region_orig %in% sample_regions) %>%
    subarray(region_dest %in% sample_regions)

p <- dplot(~ time | region_dest * region_orig,
           data = admin,
           col = "grey",
           xlab = "",
           ylab = "",
           scales = list(tck = 0.3),
           na.rm = TRUE,
           as.table = TRUE,
           midpoints = "time",
           par.settings = list(fontsize = list(text = 6),
                               strip.background = list(col = "grey90")))
p <- p + latticeExtra::layer(panel.abline(h = 1,
```

```
                                        col = "black",
                                        lty = "dotted"))
p <- useOuterStrips(p)
plot(p)
```