Bayesian Subnational Estimation using Complex Survey Data

Case Study: Kenya 2014 DHS

Zehang Richard Li

This document is an R Markdown file. It is a great way to combine data analysis with reports. To learn more about R Markdown, check out the online book at https://bookdown.org/yihui/rmarkdown/.

In this session, we will use real data from Kenya 2014 DHS survey, to illustrate spatial-temporal smoothing of child mortality rates.

Prepare data

First, we load the package and the necessary data. INLA is not in a standard repository, so we check if it is available and install it if it is not installed. For this vignette, we used INLA version 19.09.03.

```
library(SUMMER)
library(ggplot2)
library(gridExtra)
```

The DHS data can be obtained from the DHS program website at https://dhsprogram.com/data/dataset/Kenya_Standard-DHS_2014. For the analysis of U5MR, we will use the Births Recode in .dta format. Notice that registration with the DHS program is required in order to access this dataset. The map files for this DHS can be freely downloaded from http://spatialdata.dhsprogram.com/boundaries/.

With both the DHS birth record data and the corresponding shapefiles saved in the local directory. We can load them into with packages `readstata13` and `rgdal`. We also automatically generates the spatial adjacency matrix `Amat` using the function `getAmat()`.

```
library(readstata13)
filename <- "../data/KEBR71DT/KEBR71FL.DTA"
births <- read.dta13(filename, generate.factors = TRUE)

library(rgdal)
mapfilename0 <- "../data/shapefiles/sdr_subnational_boundaries.shp"
geo0 <- readOGR(mapfilename0, verbose = FALSE)
mapfilename <- "../data/shapefiles/sdr_subnational_boundaries2.shp"
geo <- readOGR(mapfilename, verbose = FALSE)
Amat <- getAmat(geo, geo$REGNAME)
```

The `Amat` matrix encodes the spatial adjacency matrix of the 47 counties, with column and row names matching the regions used in the map. This adjacency matrix will be used for the spatial smoothing model. It can also be created by hand if necessary.

Smoothed Direct Estimates

Prepare person-month data

We first demonstrate the method that smooths the direct estimates of subnational-level U5MR. For this analysis, we consider the \(8\) Admin-1 region groups. In order to calculate the direct estimates of U5MR, we need the full birth history data in the format so that every row corresponds to a birth and columns that contain:

- Indicators corresponding to survey design, e.g., strata (`v023`), cluster (`v001`), and household (`v002`)
- Survey weight (`v025`)
- Date of interview in century month codes (CMC) format, i.e., the number of the month since the beginning of 1990 (`v008`)

- Date of child's birth in CMC format (`b3`)
- Indicator for death of child (`b5`)
- Age of death of child in months (`b7`)

Since county labels are usually not in the DHS dataset, we now load the GPS location of each clusters and map them to the corresponding counties.

```
loc <- readOGR("../data//KEGE71FL/KEGE71FL.shp", verbose = FALSE)
loc.dat <- data.frame(cluster = loc$DHSCLUST, long = loc$LONGNUM, lat = loc$LATNUM)
gps <- mapPoints(loc.dat, geo = geo, long = "long", lat = "lat", names = c("REGNAME"))
colnames(gps)[4] <- "region"
gps0 <- mapPoints(loc.dat, geo = geo0, long = "long", lat = "lat", names = c("REGNAME"))
colnames(gps0)[4] <- "province"
gps <- merge(gps, gps0[, c("cluster", "province")])
sum(is.na(gps$region))
```

```
## [1] 9
```

Notice that there are \(9\) clusters that fall on the county boundaries and we need to manually assign them to a county based on best guess. In this example as an illustration, we remove these clusters without GPS coordinates.

```
unknown_cluster <- gps$cluster[which(is.na(gps$region))]
gps <- gps[gps$cluster %in% unknown_cluster == FALSE, ]
births <- births[births$v001 %in% unknown_cluster == FALSE, ]
births <- merge(births, gps[, c("cluster", "region", "province")], by.x = "v001",
    by.y = "cluster", all.x = TRUE)
births$v024 <- births$region
```

The birth history data from DHS is already in this form and the `getBirths` function default to use the current recode manual column names (as indicated above). The name of these fields can be defined explicitly in the function arguments too. We reorganize the data into the 'person-month' format with `getBirths` function and reorder the columns for better readability.

```
dat <- getBirths(data = births, strata = c("v023"), year.cut = seq(1985, 2020,
    by = 1))
dat <- dat[, c("v001", "v002", "v024", "time", "age", "v005", "strata", "died")]
colnames(dat) <- c("clustid", "id", "region", "time", "age", "weights", "strata",
    "died")
years <- levels(dat$time)
head(dat)
```

```
##   clustid id  region time  age weights strata died
## 1       1  6 nairobi 2009    0 5476381      1    0
## 2       1  6 nairobi 2009 1-11 5476381      1    0
## 3       1  6 nairobi 2009 1-11 5476381      1    0
## 4       1  6 nairobi 2009 1-11 5476381      1    0
## 5       1  6 nairobi 2009 1-11 5476381      1    0
## 6       1  6 nairobi 2009 1-11 5476381      1    0
```

Notice that we also need to specify the time intervals of interest. In this example, we with to calculate and predict U5MR in 5-year intervals from 1985-1990 to 2015-2019. For U5MR, we will use the discrete survival model to calculate direct estimates for each region and time. This step involves breaking down the age of each death into discrete intervals. The default option assumes a discrete survival model with six discrete hazards (probabilities of dying in a particular interval, given survival to the start of the interval) for each of the age bands: \([0,1), [1,12), [12,24), [24,36), [36,48)\), and \([48,60]\).

We may also calculate other types of mortality rates of interest using `getBirths`. For example, for U1MR,

```
dat_infant <- getBirths(data = births, month.cut = c(1, 12), strata = c("v023"))
```

And the smoothing steps can be similarly carried out.

Horvitz-Thompson estimators of U5MR

Using the person-month format data, we can calculate Horvitz-Thompson estimators using `getDirect` for a single survey or `getDirectList` for multiple surveys. The discrete hazards in each time interval are estimated using a logistic regression model, with weighting to account for the survey design. The direct estimates are then calculated using the discrete hazards. In order to correctly account for survey design, We need to specify the stratification and cluster variables. In the Kenya DHS example, a two-stage stratified cluster sampling design was used, where strata are specified in the `strata` column, and clusters are specified by the cluster ID (`clusterid`) and household ID (`id`).

```
direct0 <- getDirect(births = dat, years = years, regionVar = "region", timeVar = "time",
    clusterVar = "~clustid + id", ageVar = "age", weightsVar = "weights", geo.recode = NULL)
```

Adjustments using external information

Sometimes additional information are available to adjust the direct estimates from the surveys. For example, in countries with high prevalence of HIV, estimates of U5MR can be biased, particularly before ART treatment became widely available. Pre-treatment HIV positive women had a high risk of dying, and such women who had given birth were therefore less likely to appear in surveys. The children of HIV positive women are also more likely to have a higher probability of dying compared to those born to HIV negative women. Thus we expect that the U5MR is underestimated if we do not adjust for the missing women.

Suppose we can obtain the ratio of the reported U5MR to the true U5MR, r_{it}, at region i and time period t, we can apply the adjustment factor to the direct estimates and the associated variances. The HIV adjustment factors were calculated for the 2014 Kenya DHS survey and included in the package.

```
data(KenData)
adj <- KenData$HIV2014.yearly
colnames(adj) <- c("years", "province", "ratio")
head(adj)
```

```
##   years province ratio
## 1  2014      All  0.99
## 2  2013      All  0.98
## 3  2012      All  0.98
## 4  2011      All  0.97
## 5  2010      All  0.97
## 6  2009      All  0.96
```

The `KenData$HIV2014.yearly` data frame contains HIV adjustment factors at both national and province levels. So we will create another column in `direct0` that codes the province each county belongs to.

```
matched <- match(direct0$region, gps$region)
direct0$province <- as.character(gps[matched, "province"])
direct0$province[direct0$region == "All"] <- "All"
direct0$logit.lower <- logit(direct0$lower)
direct0$logit.upper <- logit(direct0$upper)
direct <- getAdjusted(data = direct0, ratio = adj, time = "years", region = "province",
    adj = "ratio", logit.lower = "logit.lower", logit.upper = "logit.upper",
    lower = "lower", upper = "upper")
```

National estimates of U5MR

The direct estimates calculated using `getDirect` contains both national and subnational estimates for the 47 regions, over the 35 years from 1985 to 2019. We first fit a model with temporal random effects only to smooth the national estimates over time. In this part, we use the subset of data region variable being “All”.

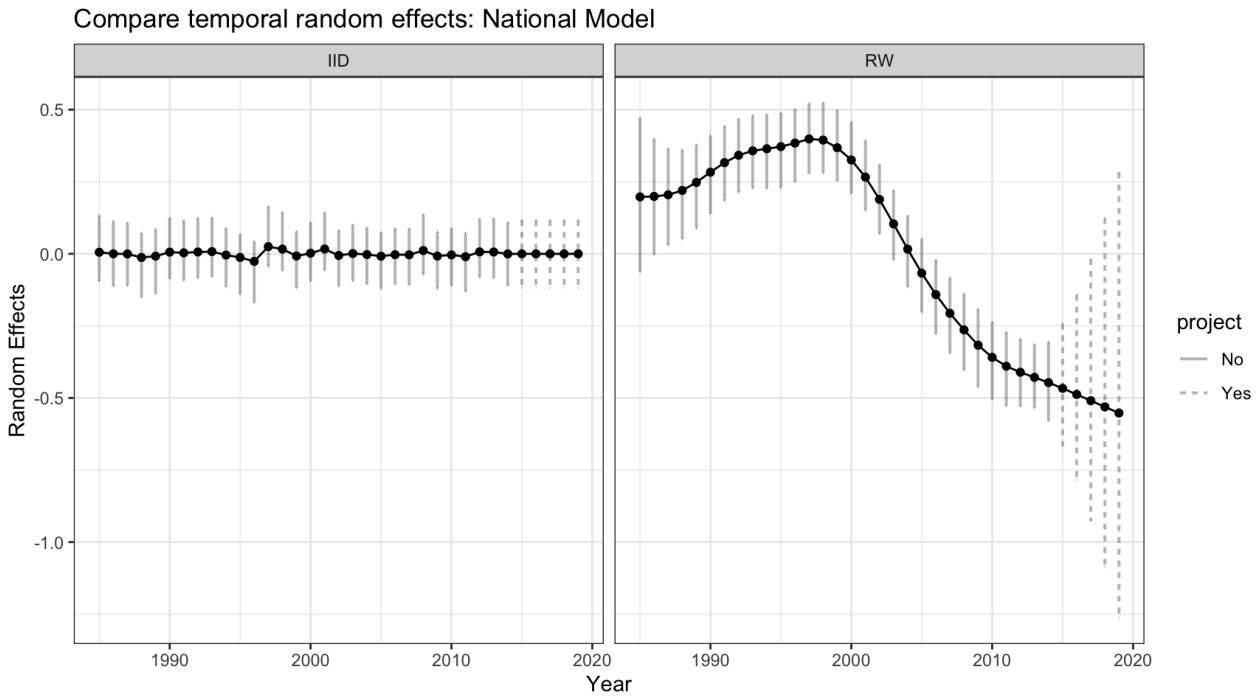
```
fit0 <- fitINLA(data = direct, geo = NULL, Amat = NULL, year_label = years,
    year_range = c(1985, 2019), rw = 2, m = 1)
```

The marginal posteriors are already stored in the fitted object. We use the following function to extract and re-arrange them.

```
out0 <- getSmoothed(fit0, year_range = c(1985, 2019), year_label = years)
```

We can compare the results visually. Notice to correctly display the period estimates, the reference year in each period needs to be specified. Here we simply take the median year in each period.

```
random.time <- getDiag(fit0, field = "time", year_label = years)
plot(random.time, is.subnational = FALSE) + facet_grid(~label) + ggtitle("Compare temporal random e
ffects: National Model") +
    ylab("Random Effects")
```



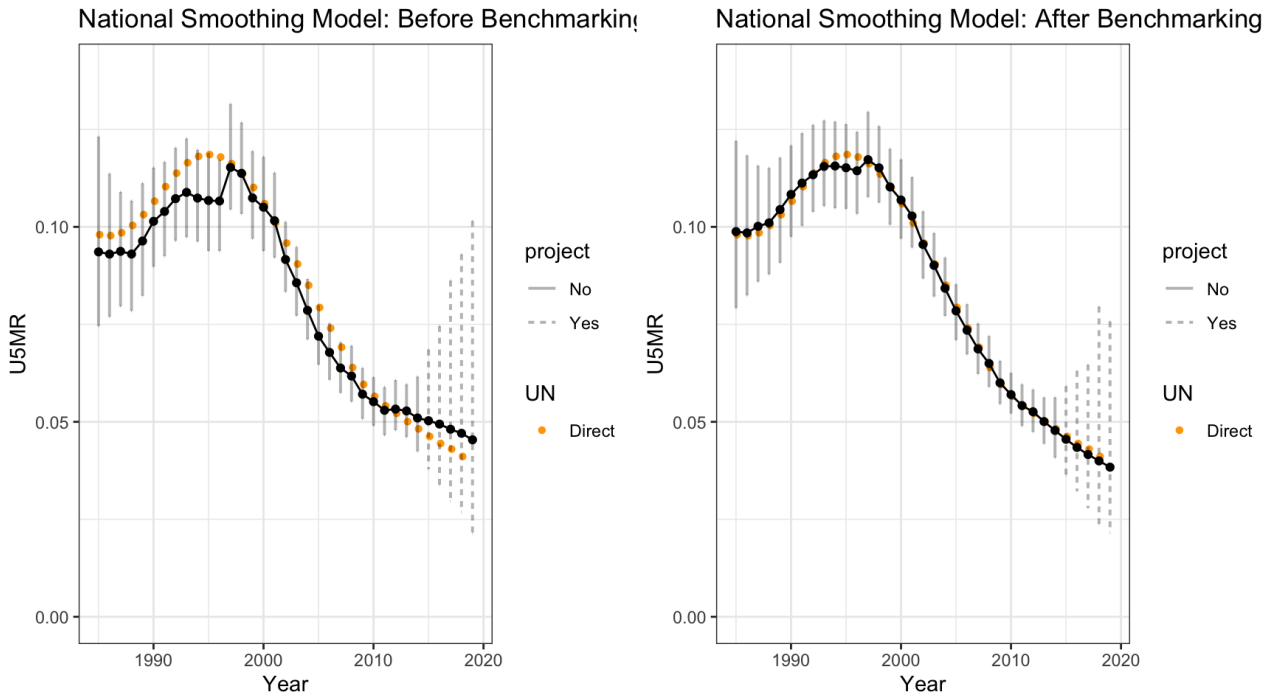
The national model also allows us to benchmark the estimates using other published national results. For example, we take the 2019 UN-IGME estimates and calculate the ratio of the estimates from national models to the published UN estimates. We will use this adjustment ratio to correct the bias from our direct estimates. We organize the adjustment ratios into a matrix of two columns, since the adjustment factor only varies over time. We can then perform the benchmarking to the UN estimates similar to the HIV adjustment before.

```
UN <- KenData$IGME2019
ratio <- out0$median[1:34]/UN$mean[34:67]
adj.benchmark <- data.frame(years = out0$years[1:34], ratio = ratio)
direct <- getAdjusted(data = direct, ratio = adj.benchmark, time = "years",
    region = "province", adj = "ratio", logit.lower = "logit.lower", logit.upper = "logit.upper",
    lower = "lower", upper = "upper")
```

After benchmarking, we can fit the smoothing model again on the adjusted direct estimates, and see if they align with the UN estimates.

```
fit0.benchmark <- fitINLA(data = direct, geo = NULL, Amat = NULL, year_label = years,
    year_range = c(1985, 2019), rw = 2, m = 1)
out0.benchmark <- getSmoothed(fit0.benchmark, year_range = c(1985, 2019), year_label = years)

g1 <- plot(out0, year_label = years, year_med = as.numeric(years), is.subnational = FALSE,
    data.add = UN, option.add = list(point = "mean"), label.add = "UN", color.add = "orange") +
    ggtitle("National Smoothing Model: Before Benchmarking") + ylim(c(0, 0.14))
g2 <- plot(out0.benchmark, year_label = years, year_med = as.numeric(years),
    is.subnational = FALSE, data.add = UN, option.add = list(point = "mean"),
    label.add = "UN", color.add = "orange") + ggtitle("National Smoothing Model: After Benchmarking
") +
    ylim(c(0, 0.14))
grid.arrange(g1, g2, ncol = 2)
```



Subnational estimates of U5MR

The syntax to fit subnational smoothing model is similar to the national model.

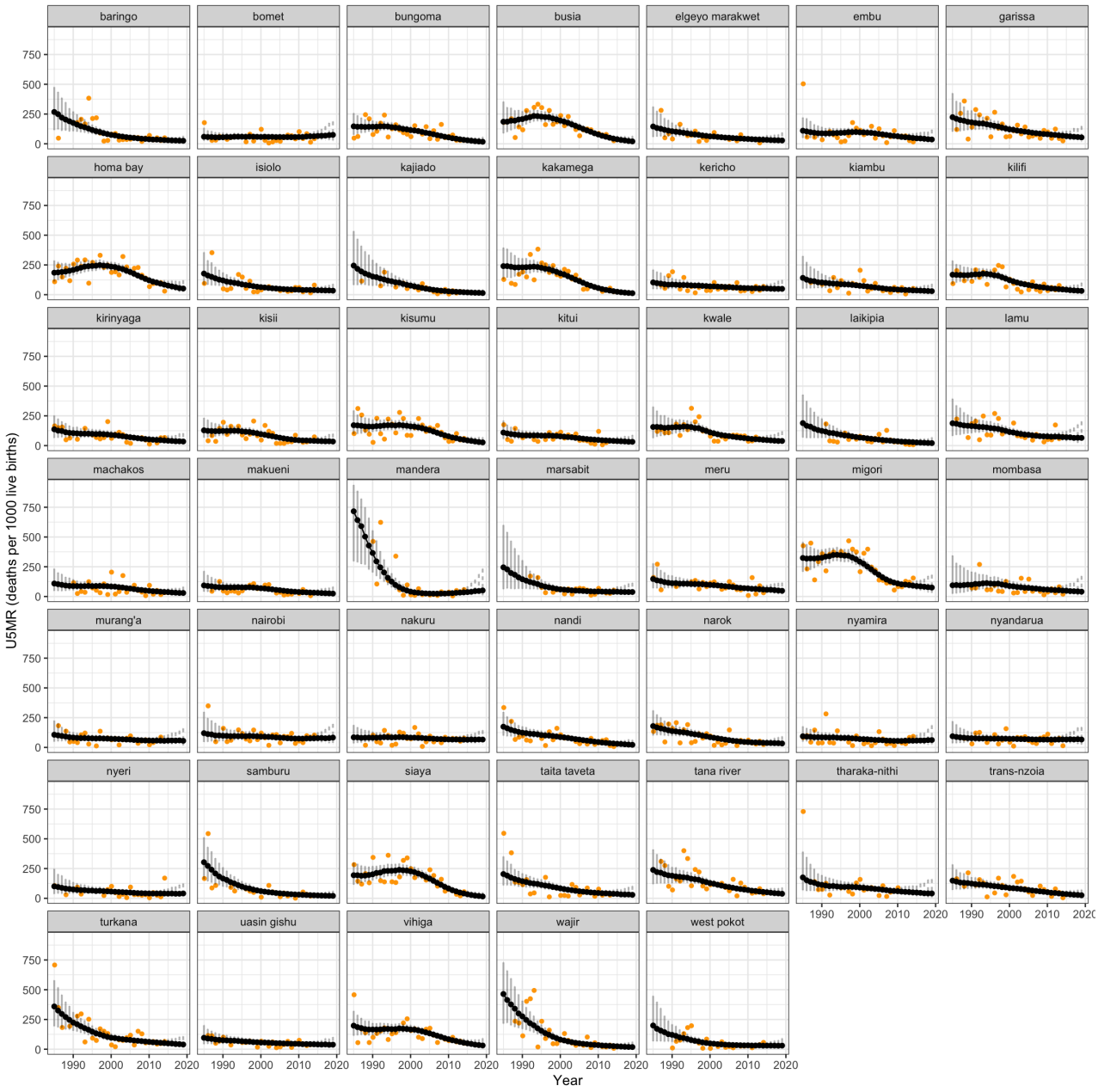
```
fit0.sub <- fitINLA(data = direct, geo = geo, Amat = Amat, year_label = years,
    year_range = c(1985, 2019), rw = 2, type.st = 4, m = 1)
```

The smoothed estimates can be

```
out0.sub <- getSmoothed(fit0.sub, Amat = Amat, year_range = c(1985, 2019), year_label = years)
```

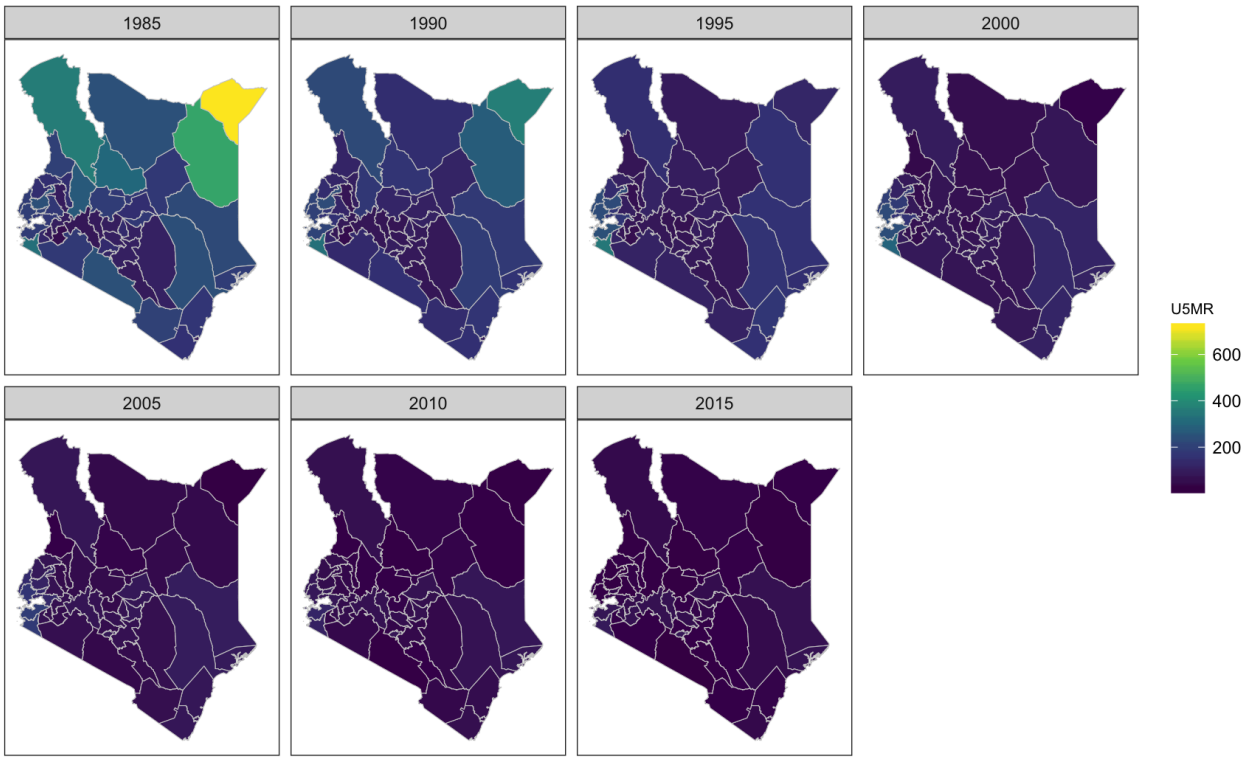
We can also add back the direct estimates for comparison.

```
plot(out0.sub, is.subnational = FALSE, data.add = direct, option.add = list(point = "mean",
    by = "survey"), color.add = "orange", per1000 = TRUE) + facet_wrap(~region,
    ncol = 7) + theme(legend.position = "none") + scale_color_manual(values = rep("gray20",
    47))
```



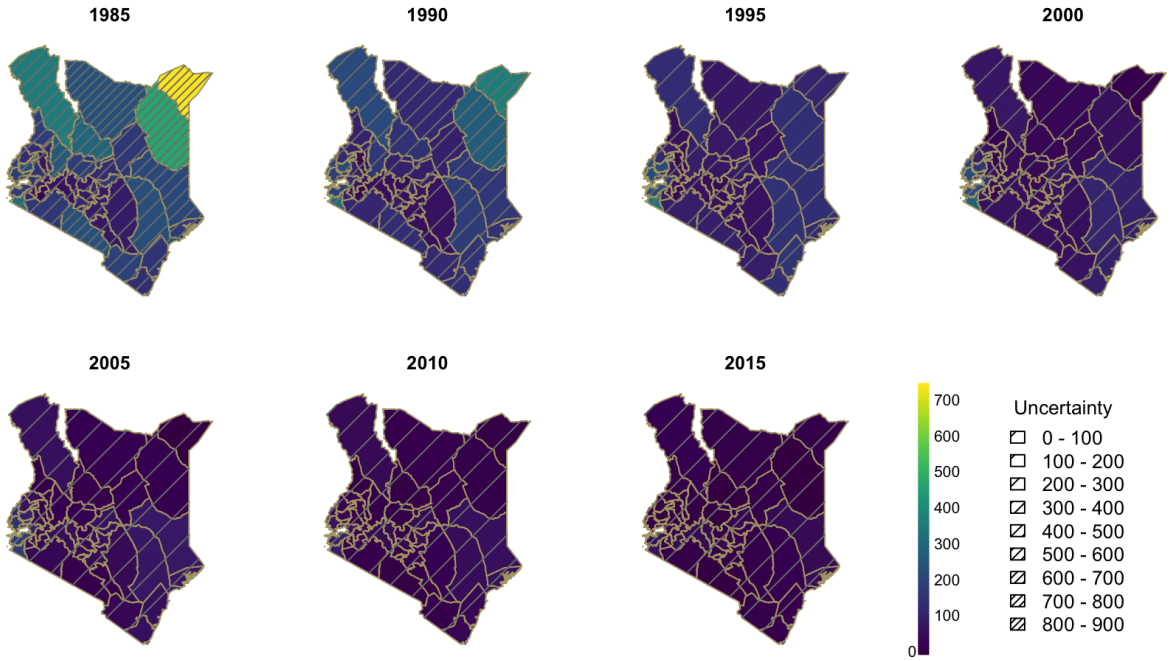
We can show the estimates over time on maps.

```
mapPlot(data = subset(out0.sub, years %in% c(1985, 1990, 1995, 2000, 2005, 2010,
    2015)), geo = geo, variables = c("years"), values = c("median"), by.data = "region",
    by.geo = "REGNAME", is.long = TRUE, border = "gray80", size = 0.2, ncol = 4,
    per1000 = TRUE, legend.label = "U5MR")
```



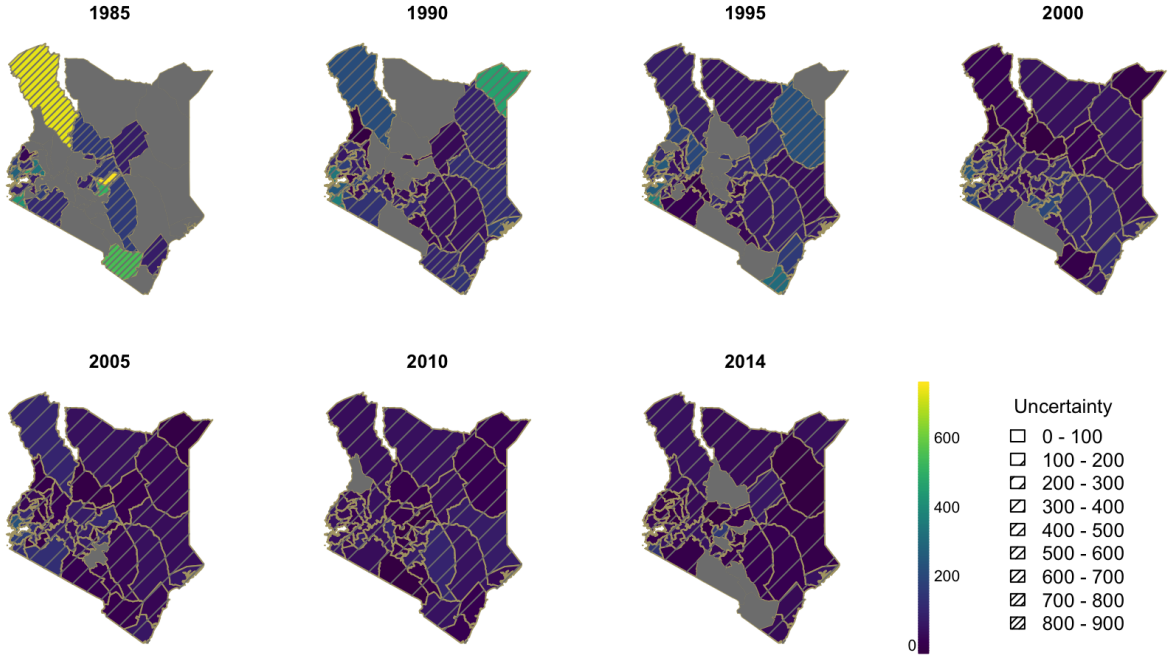
In order to also illustrate uncertainties of the estimates for some selected years when presented on maps, we can use hatching to indicate the width of the 95% posterior credible intervals.

```
breaks.hatch <- seq(0, 900, len = 10)
hatchPlot(data = subset(out0.sub, years %in% c(1985, 1990, 1995, 2000, 2005,
    2010, 2015)), geo = geo, variables = c("years"), values = c("median"), by.data = "region",
    by.geo = "REGNAME", lower = "lower", upper = "upper", is.long = TRUE, ncol = 4,
    hatch = "gray50", per1000 = TRUE, breaks.CI = breaks.hatch)
```



We can also compare these estimates to the direct estimates. Since we have adjusted the logits of the direct estimates, we will calculate the corresponding confidence intervals for the adjusted direct estimates first.

```
hatchPlot(data = subset(direct, years %in% c(1985, 1990, 1995, 2000, 2005, 2010,
    2014)), geo = geo, variables = c("years"), values = c("mean"), by.data = "region",
    by.geo = "REGNAME", lower = "lower", upper = "upper", is.long = TRUE, ncol = 4,
    hatch = "gray50", per1000 = TRUE, breaks.CI = breaks.hatch)
```

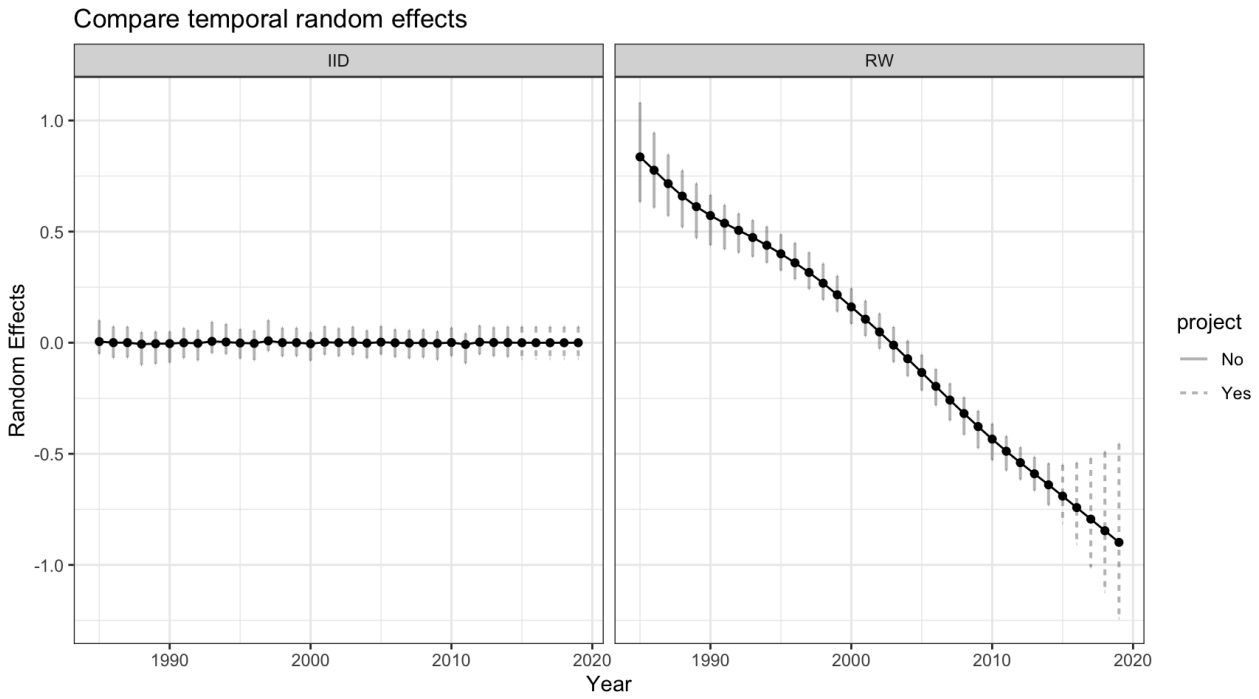


Diagnostics

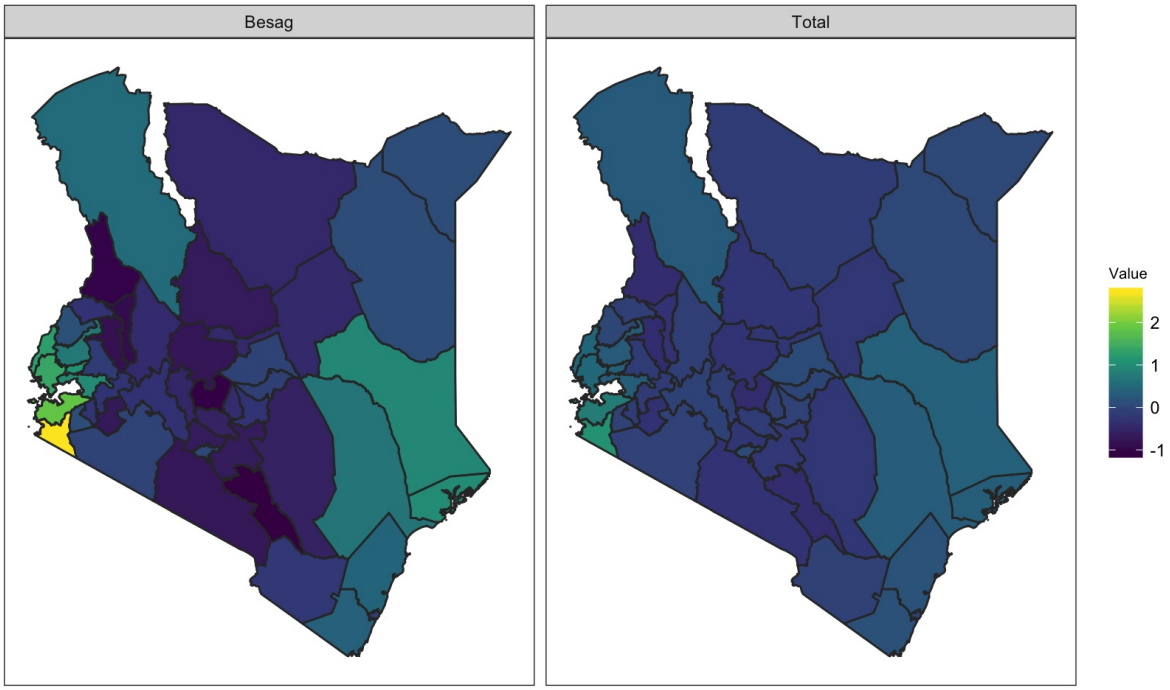
We can extract the spatial, temporal, and spatial-temporal random effects using the `getDiag` function.

```
random.time <- getDiag(fit0.sub, field = "time", year_label = years)
random.space <- getDiag(fit0.sub, field = "space", Amat = Amat)
random.spacetime <- getDiag(fit0.sub, field = "spacetime", year_label = years,
    Amat = Amat)
```

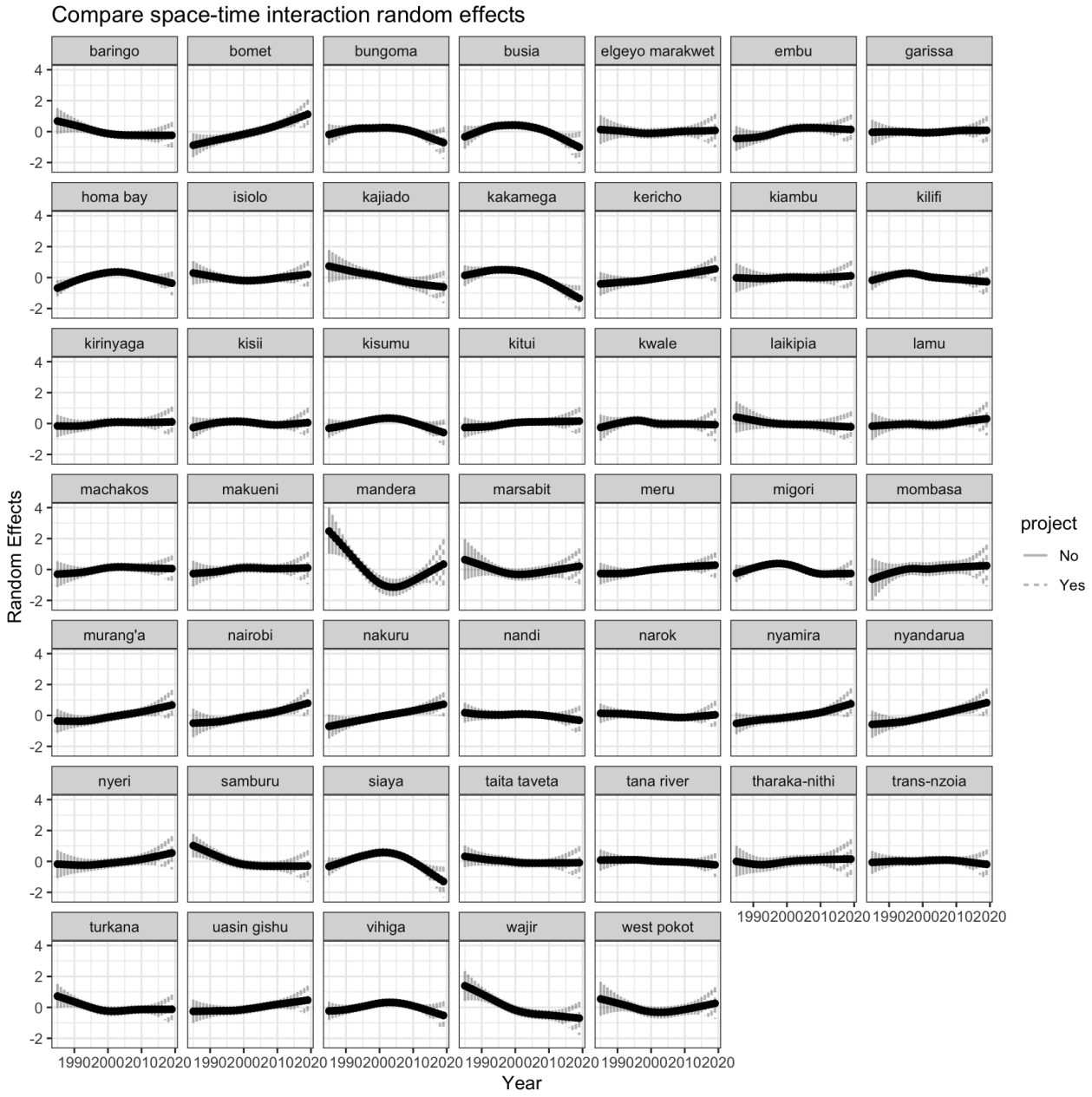
```
plot(random.time, is.subnational = FALSE) + facet_wrap(~label) + ggtitle("Compare temporal random e
ffects") +
    ylab("Random Effects")
```




```
mapPlot(random.space, geo = geo, by.data = "region", by.geo = "REGNAME", variables = "label",
    values = c("median"), ncol = 2, is.long = TRUE)
```




```
plot(random.spacetime, is.subnational = FALSE) + facet_wrap(~region, ncol = 7) +
    ylab("Random Effects") + ggtitle("Compare space-time interaction random effects")
```



Model-based Estimates

Assume there are N regions and T time periods. In the i-th region, there are n_i clusters. We consider the following model. In stratum k, cluster c, and time period t, let $p_{ktc}^{(g)}$ be the probability of a death in age group g conditional on surviving until the end of the previous age group, i.e., the hazard of age group g. Given $p_{ktc}^{(g)}$, let n_g denote the number of months in age group g, the under-5 mortality rate in stratum k, cluster c, time period t is $$ p_{ktc} = 1 - \prod_g (1 - p_{ktc}^{(g)})^{n_g} $$ and further aggregating over all clusters within each region, $$ p_{it} = \sum_{k}\sum_{c} p_{ktc} q_{ik} \mathbf{1}_{i[c] = i}, $$ where q_{ik} is the proportion of clusters that are in stratum k within region i. Let $Y_{ktc}^{(g)}$ and $n_{ktc}^{(g)}$ to denote the number of deaths and the total number of child-months in stratum k, cluster c, time period t, and age group g. We model the hazard $p_{ktc}^{(g)}$ with a hierarchical space-time smoothing model described below.

Beta-binomial model

We assume the following marginal model, $$ Y_{ktc}^{(g)} | p_{ktc}^{(g)} \sim \mbox{BetaBinomial}(n_{ktc}^{(g)}, p_{ktc}^{(g)}, d). $$ We model the mean probability $p_{ktc}^{(g)}$ with a logit link and a linear model that contains strata and age group fixed effects, and space, time, and space-time random effects $$ \mbox{logit} p_{ktc}^{(g)} = \log(\mbox{BIAS}_{tc}) + \mu_g + \beta_k + \alpha_t^{(g)} + \gamma_t + \phi_{i[c]} + \delta_{i[c],t} , $$ where the bias term $\mbox{BIAS}_{tc}$ denotes the ratio of the reported U5MR to the 'true' U5MR. The log transformation on the logit transformed hazards approximately leads to a multiplicative bias correction on the U5MR. In countries with high prevalence of HIV, we may adjust for the proportion of missing women due to HIV prevalence. And the random effects are defined similarly as in the previous smoothing model. In our example, we let the first two age groups, $[0, 1)$ and $[1, 11)$ months, each have a set of distinct random walk effects, and let the rest of the age groups share the same random walk effects. We let the three groups of random walks share the same smoothing parameter.

We now transform the full birth history data to person-month format again. In order to fit the binomial model, we need to

calculate the number of person-months and number of deaths for each age group, stratum (urban or rural), cluster, and time period. Notice that we do not need to impute 0 observations for future time periods. We also rename the columns to prepare for the input in the smoothing model. In order to correctly adjust for bias due to HIV, we keep the information of province in the column 'province' as well.

```
dat <- getBirths(data = births, strata = c("v023"), year.cut = seq(1985, 2020,
    by = 1))
dat <- dat[, c("v001", "time", "age", "died", "v025", "v024")]
colnames(dat) <- c("cluster", "years", "age", "Y", "strata", "province")
outcome <- getCounts(data = dat, variables = "Y", by = c("age", "strata", "cluster",
    "years"), ignore = list(years = c(2015:2019)))
head(outcome)
```

```
##     age strata cluster years Y total
## 1     0  urban       1  1985 0     0
## 2  1-11  urban       1  1985 0     0
## 3 12-23  urban       1  1985 0     0
## 4 24-35  urban       1  1985 0     0
## 5 36-47  urban       1  1985 0     0
## 6 48-59  urban       1  1985 0     0
```

We then merge the county information to this dataset. In order to fit the model, the data file should contain the following columns: cluster ID ('cluster'), observation period ('years'), observation location ('region'), strata level ('strata'), age group corresponding to the hazards ('age'), total number of person-months ('total'), and the number of deaths ('Y').

```
outcome <- merge(outcome, gps[, c("cluster", "region", "province")], by = "cluster",
    all.x = TRUE)
```

We will use the province level HIV adjustment factors to adjust our observed hazards. In the `KenData2014` data included in the package, we calculated ratios of the reported U5MR to the true U5MR, r_{it}, at province i and time period t. We can apply the adjustment factor to the hazards. In order for the two datasets to be automatically merged, we changed the column name in the HIV adjustment ratio data to match the one in `outcome`. We then fit the binomial space-time smoothing model to obtain the yearly estimates of U5MR.

```
adj <- KenData$HIV2014.yearly
colnames(adj) <- c("years", "province", "ratio")
```

Benchmarking using national-level model

In order to benchmark the estimates to other published results at the national model, we first fit the following simplified binomial model to the data to obtain estimates at the national level. We again model the hazards using a (time-only) smoothing model: $$ \mbox{logit} p_{ktc}^{(g)} = \log(\mbox{BIAS}_{tc}) + \mu_g + \beta_k + \alpha_t + \gamma_t , $$ Notice that for the national model, instead of using $k = 2$ (urban/rural) strata, we use the actual stratification variable in the data, which is usually the interaction of region and urban/rural. We then stratum-specific estimates the results using the proportion of population in each stratum. In this case study, we take the 2019 UN-IGME yearly estimates and calculate the ratio of the estimates from national models to the published UN estimates. Then we can update the bias adjustment ratio to be $$ \mbox{BIAS}_{tc}^{\mbox{(new)}} = \mbox{BIAS}_{tc} \times \frac{\widehat{\mbox{U5MR}}_{t}}{\mbox{U5MR}^{\mbox{(UN)}}_{t}} $$

National level model

We first fit the national-level model to the data. We create a new data frame with strata defined as the cross interaction of the 47 counties and the urban/rural stratification.

```
outcome_national <- outcome
outcome_national$strata <- paste(outcome$region, outcome$strata)
```

As described before, we let the first two age groups, $[0, 1)$ and $[1, 11)$ months, each have a set of distinct random walk effects, and let the rest of the age groups share the same random walk effects. This is specified using the argument `age.rw.group`.

```
fit1 <- fitINLA2(data = outcome_national, family = "betabinomial", Amat = NULL,
    geo = NULL, year_label = 1985:2019, bias.adj = adj, bias.adj.by = c("years",
        "province"), age.rw.group = c(1, 2, 3, 3, 3, 3), verbose = FALSE)
```

In order to calculate the smoothed national estimates, we need the proportion of population in each of the 92 strata in the Kenya 2014 DHS. For most DHS surveys, we can obtain such information from the DHS final reports. We have included the table for 2014 Kenya DHS in the package in `KenData$UrbanProp`. The urban/rural proportions are calculated for each county in this table, so we need to further calculate the proportion of population in each strata.

```
Prop <- KenData$UrbanProp
Prop_national <- data.frame(matrix(NA, 1, dim(Amat)[1] * 2))
colnames(Prop_national) <- c(paste(Prop$region, "urban"), paste(Prop$region,
    "rural"))
Prop_national[1, ] <- c(Prop$urban * Prop$population, Prop$rural * Prop$population)
Prop_national[1, ] <- Prop_national[1, ]/sum(Prop_national[1, ])
```

We then simulate from the posterior to obtain the estimates of U5MR. Notice that for binomial model, `getSmoothed` returns a list of two objects, one for the overall estimates for each region and time, and another for stratum-specific estimates.

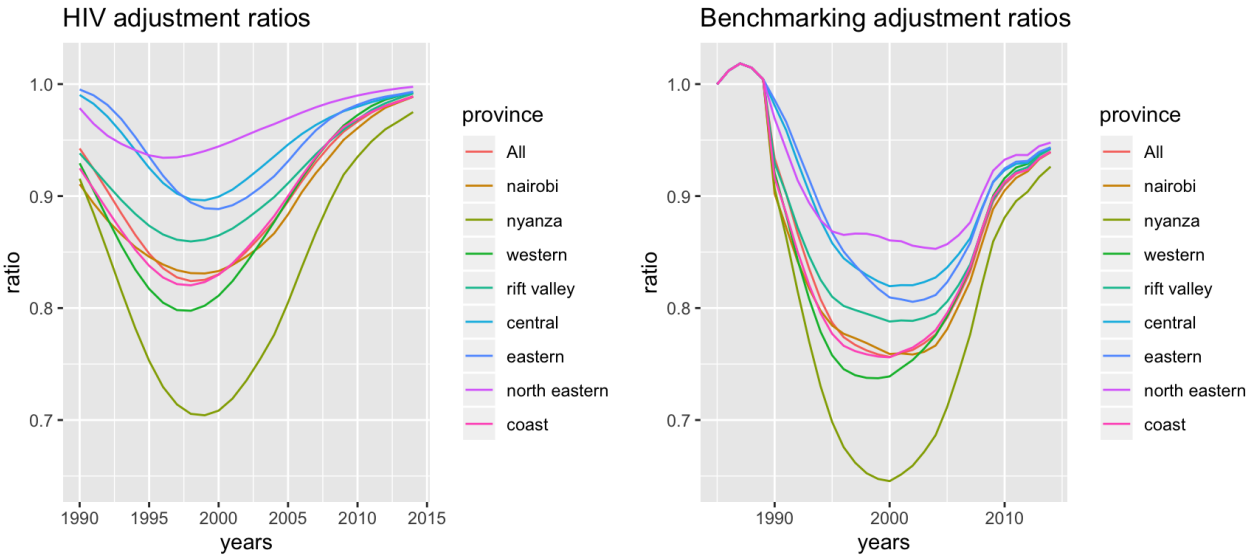
```
out1 <- getSmoothed(inla_mod = fit1, year_label = years, Amat = Amat, nsim = 1000,
    weight.strata = Prop_national)
```

Using the yearly smoothed estimates, we can calculate a new set of bias adjustment factors.

```
data(KenData)
UN <- KenData$IGME2019
ratio <- out1$overall$median[1:34]/UN$mean[34:67]
adj.benchmark <- expand.grid(years = 2014:1985, province = unique(adj$province))
adj.benchmark <- merge(adj.benchmark, adj, all.x = TRUE)
adj.benchmark$ratio[is.na(adj.benchmark$ratio)] <- 1
adj.benchmark$ratio <- adj.benchmark$ratio * ratio[adj.benchmark$years - 1984]
```

We visually compare the new adjustment ratios with the original HIV adjustment ratios.

```
g1 <- ggplot(data = subset(adj, years >= 1985), aes(x = years, y = ratio, group = province,
    color = province)) + geom_line() + ylim(range(adj.benchmark$ratio)) + ggtitle("HIV adjustment r
atios")
g2 <- ggplot(data = adj.benchmark, aes(x = years, y = ratio, group = province,
    color = province)) + geom_line() + ggtitle("Benchmarking adjustment ratios")
grid.arrange(g1, g2, ncol = 2)
```

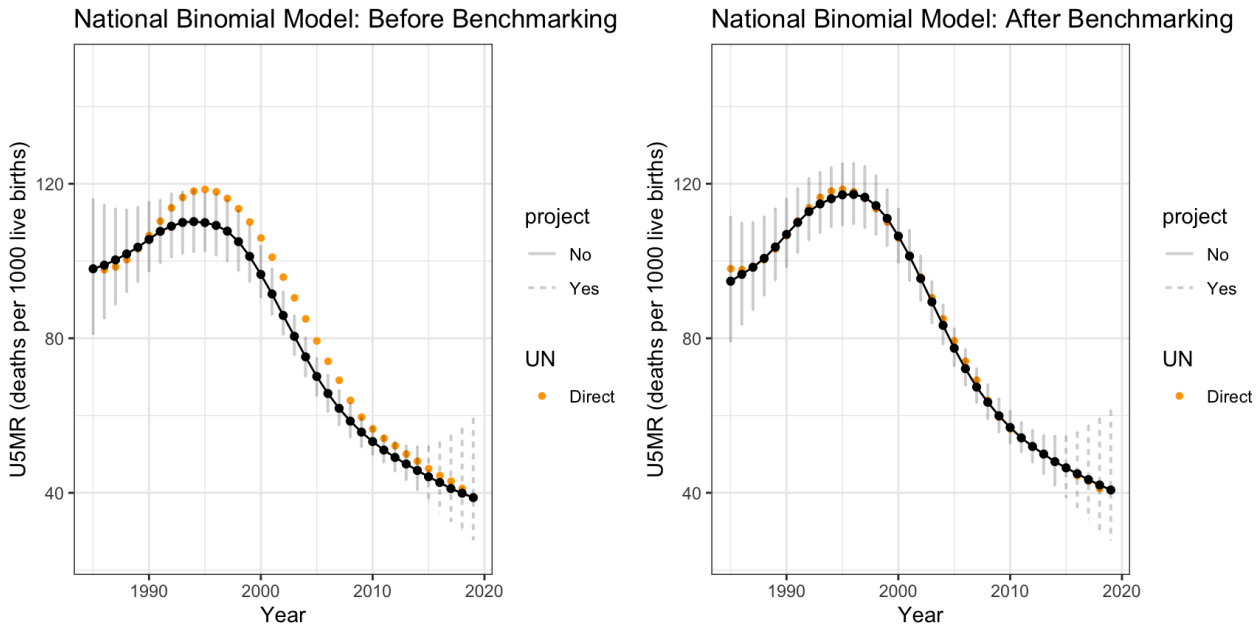


Using the benchmarked adjustment ratio, we may refit the national model and compare the results.

```
fit1.benchmark <- fitINLA2(data = outcome_national, family = "betabinomial",
    Amat = NULL, geo = NULL, year_label = 1985:2019, bias.adj = adj.benchmark,
    bias.adj.by = c("years", "province"), age.rw.group = c(1, 2, 3, 3, 3, 3),
    verbose = FALSE)
out1.benchmark <- getSmoothed(inla_mod = fit1.benchmark, year_label = years,
    Amat = NULL, nsim = 1000, weight.strata = Prop_national)
```

We can compare the national estimates before and after benchmarking.

```
g1 <- plot(out1$overall, year_label = years, year_med = 1985:2019, is.subnational = FALSE,
    plot.CI = TRUE, alpha.CI = 0.2, per1000 = TRUE, data.add = UN, option.add = list(point = "mean"
),
    label.add = "UN", color.add = "orange", dodge.width = 0.1) + ggtitle("National Binomial Model:
Before Benchmarking") +
    ylim(25, 150)
g2 <- plot(out1.benchmark$overall, year_label = years, year_med = 1985:2019,
    is.subnational = FALSE, plot.CI = TRUE, alpha.CI = 0.2, per1000 = TRUE,
    data.add = UN, option.add = list(point = "mean"), label.add = "UN", color.add = "orange",
    dodge.width = 0.1) + ggtitle("National Binomial Model: After Benchmarking") +
    ylim(25, 150)
grid.arrange(g1, g2, ncol = 2)
```

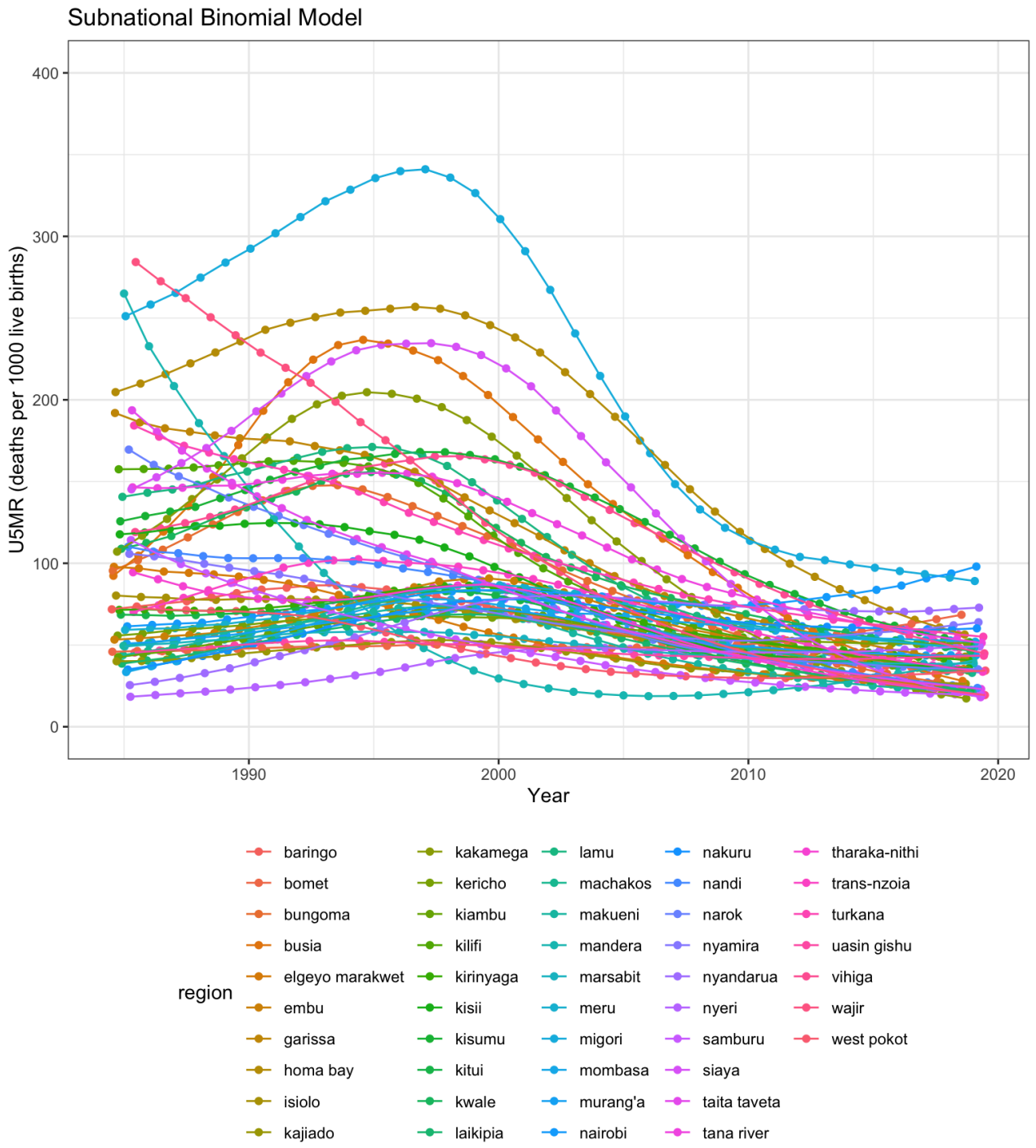


Using the benchmarked adjustment ratio, we now fit the full model and calculate the estimates and their 95% posterior credible intervals using \(1,000\) draws from the posterior distribution.

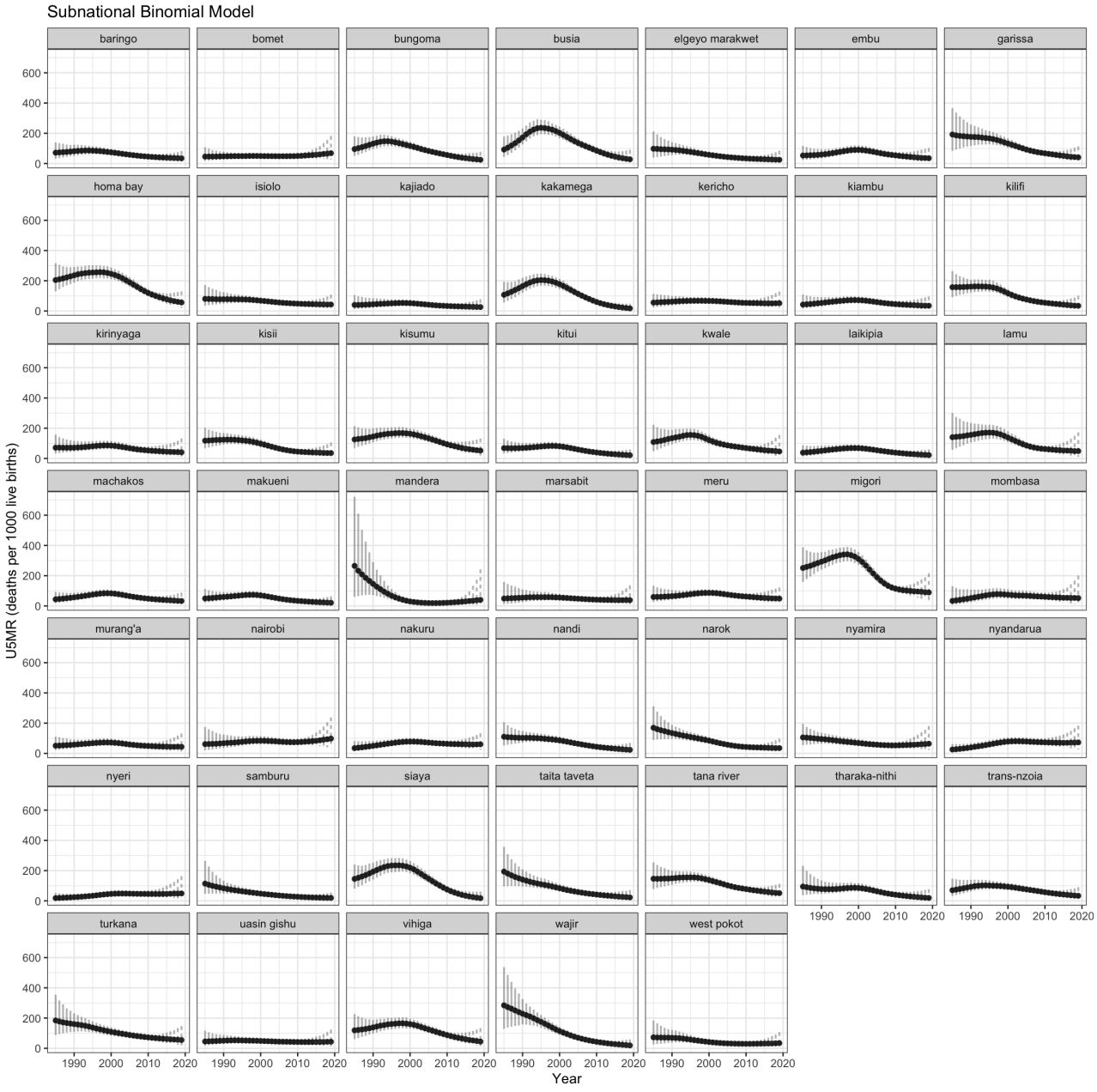
```
fit1.sub <- fitINLA2(data = outcome, family = "betabinomial", Amat = Amat, geo = geo,
    year_label = 1985:2019, type.st = 4, bias.adj = adj.benchmark, bias.adj.by = c("years",
        "province"), age.rw.group = c(1, 2, 3, 3, 3, 3), verbose = FALSE)
out1.sub <- getSmoothed(inla_mod = fit1.sub, year_label = years, Amat = Amat,
    nsim = 1000, weight.strata = KenData$UrbanProp, num.threads = 12)
```

The posterior medians of U5MR in each counties can be plotted using the `plot` method, `mapPlot` function and `hatchPlot` function.

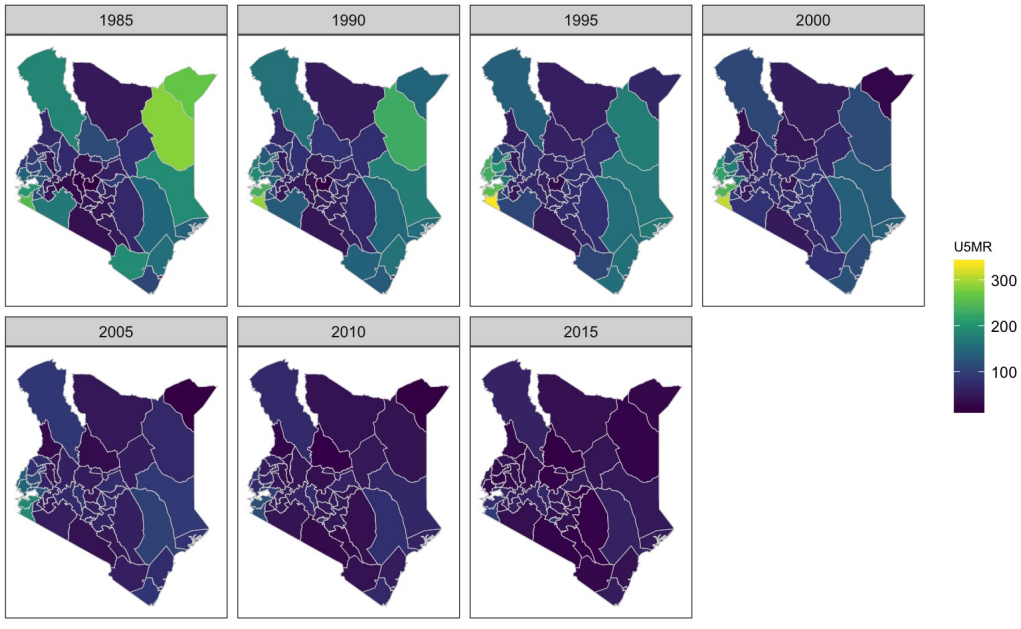
```
plot(out1.sub$overall, year_label = years, year_med = 1985:2019, is.subnational = TRUE,
    plot.CI = FALSE, per1000 = TRUE) + ggtitle("Subnational Binomial Model") +
    theme(legend.position = "bottom") + ylim(c(0, 400))
```



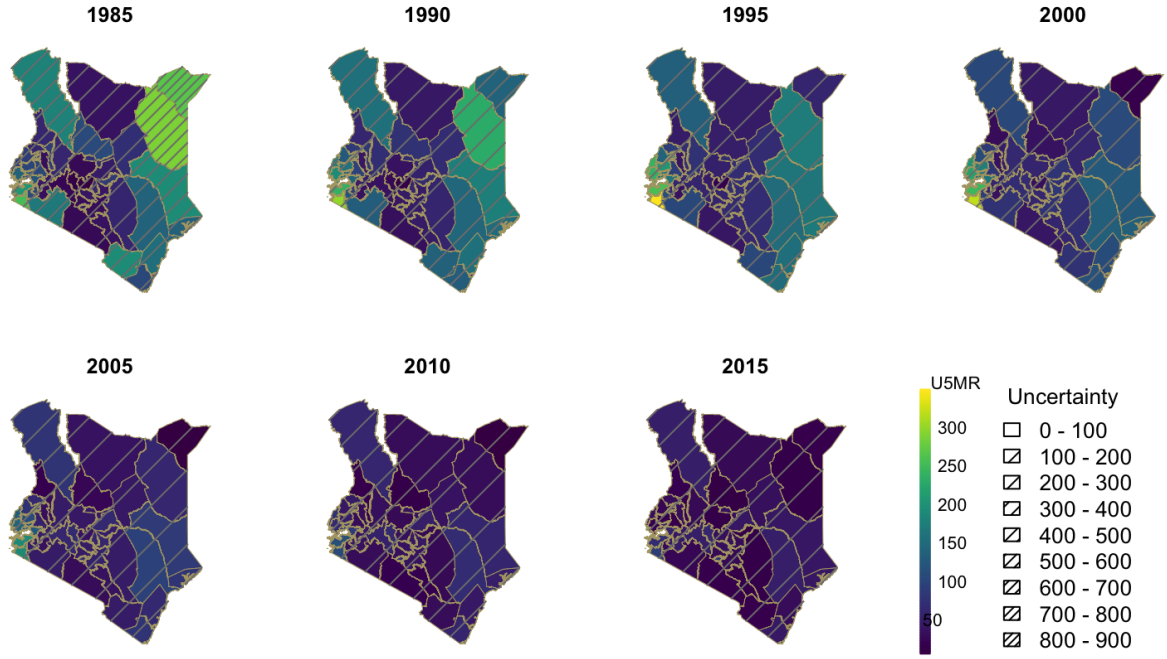

```
plot(out1.sub$overall, year_label = years, year_med = 1985:2019, is.subnational = TRUE,
    plot.CI = TRUE, per1000 = TRUE) + ggtitle("Subnational Binomial Model") +
    facet_wrap(~region, ncol = 7) + theme(legend.position = "none") + scale_color_manual(values =
rep("gray20",
    47))
```




```
mapPlot(data = subset(out1.sub$overall, years %in% c(1985, 1990, 1995, 2000,
    2005, 2010, 2015)), geo = geo, variables = c("years"), values = c("median"),
    by.data = "region", by.geo = "REGNAME", is.long = TRUE, border = "gray80",
    size = 0.2, ncol = 4, per1000 = TRUE, legend.label = "U5MR")
```



```
hatchPlot(data = subset(out1.sub$overall, years %in% c(1985, 1990, 1995, 2000,
    2005, 2010, 2015)), geo = geo, variables = c("years"), values = c("median"),
    by.data = "region", by.geo = "REGNAME", lower = "lower", upper = "upper",
    is.long = TRUE, size = 0.5, ncol = 4, per1000 = TRUE, legend.label = "U5MR",
    breaks.CI = breaks.hatch, hatch = "gray50")
```

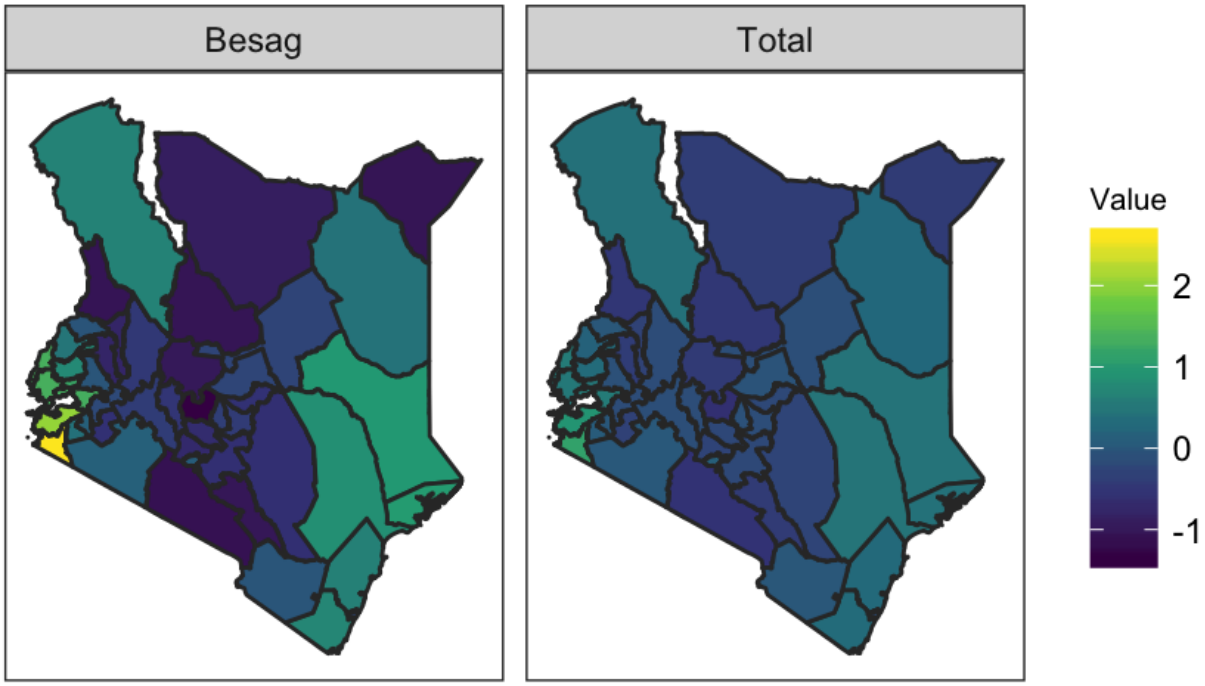


Diagnostics

Besides evaluating the model components using `summary(fit1$fit)`, we may also plot the posteriors of the random effects.

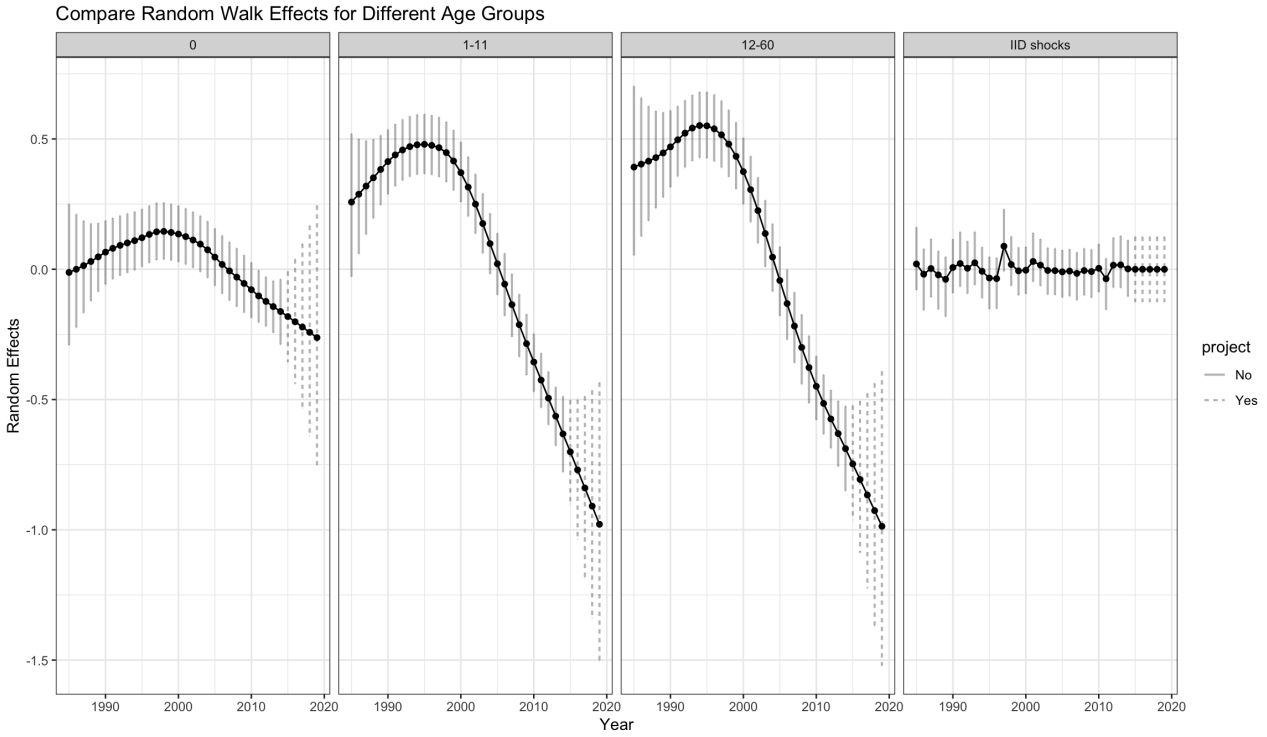
We first extract the spatial random effects, \(\phi_i\), and plots its posterior medians on the map.

```
random.space <- getDiag(fit1.sub, field = "space", Amat = Amat)
mapPlot(random.space, geo = geo, by.data = "region", by.geo = "REGNAME", variables = "label",
    values = c("median"), ncol = 2, is.long = TRUE)
```



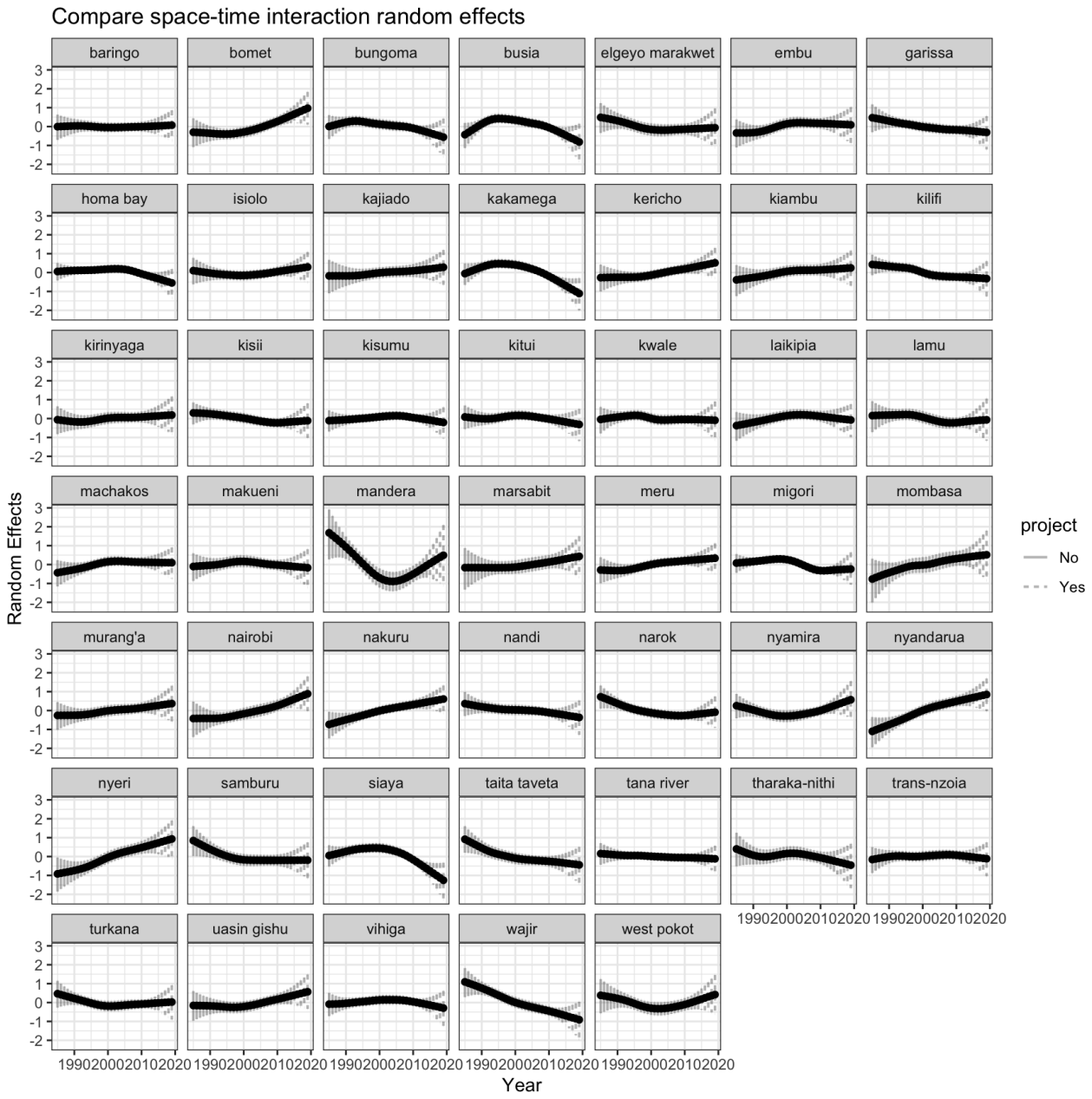
For the temporal random effects, since we model the last \(4\) age groups using the same random effects, we rename the extracted temporal random effects slightly and show them in the following figure.

```
random.time <- getDiag(fit1.sub, field = "time", year_label = years)
random.time[is.na(random.time$group), "group"] <- "IID shocks"
random.time <- subset(random.time, group %in% c("0", "1-11", "12-23", "IID shocks"))
random.time[random.time$group == "12-23", "group"] <- "12-60"
plot(random.time, is.subnational = FALSE) + facet_wrap(~group, ncol = 4) + ggtitle("Compare Random
Walk Effects for Different Age Groups") +
    ylab("Random Effects")
```



For the space-time random effects,

```
random.spacetime <- getDiag(fit1.sub, field = "spacetime", year_label = years,
    Amat = Amat)
plot(random.spacetime, is.subnational = FALSE) + facet_wrap(~region, ncol = 7) +
    ylab("Random Effects") + ggtitle("Compare space-time interaction random effects")
```



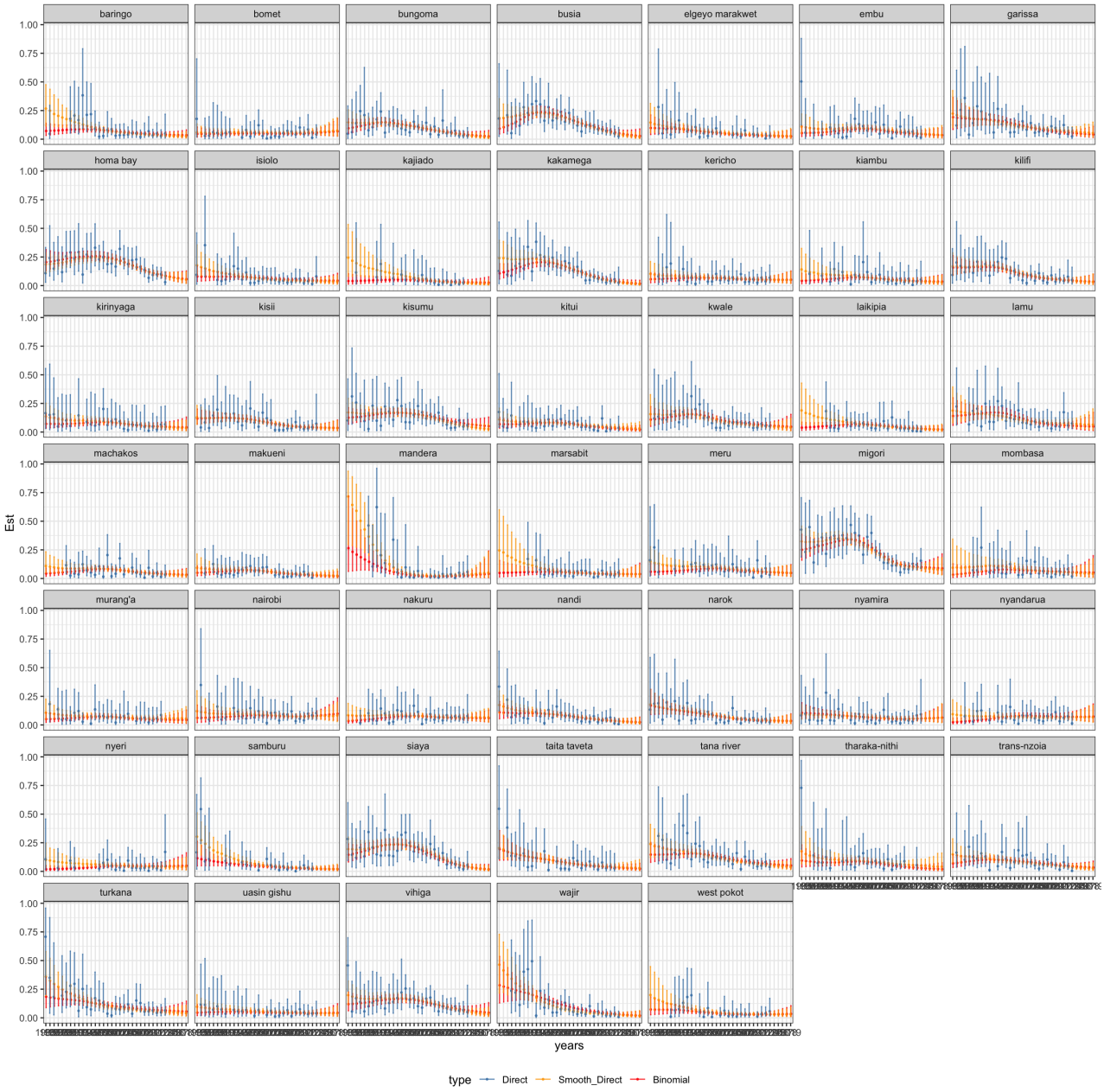
Comparisons

In this last section, we combine the results from all three estimates: the direct estimates, the smoothed direct estimates, and the model-based binomial model estimates.

```
results0 <- out0.sub[, c("region", "years", "median", "upper", "lower")]
results1 <- out1.sub$overall[, c("region", "years", "median", "upper", "lower")]
results2 <- direct[, c("region", "years", "mean", "upper", "lower")]
results0$type <- "Smooth_Direct"
results1$type <- "Binomial"
results2$type <- "Direct"
colnames(results0)[3] <- colnames(results1)[3] <- colnames(results2)[3] <- "Est"
results <- rbind(results0, results1, results2)
results$type <- factor(results$type, levels = c("Direct", "Smooth_Direct", "Binomial"))
results <- results[results$region != "All", ]
```

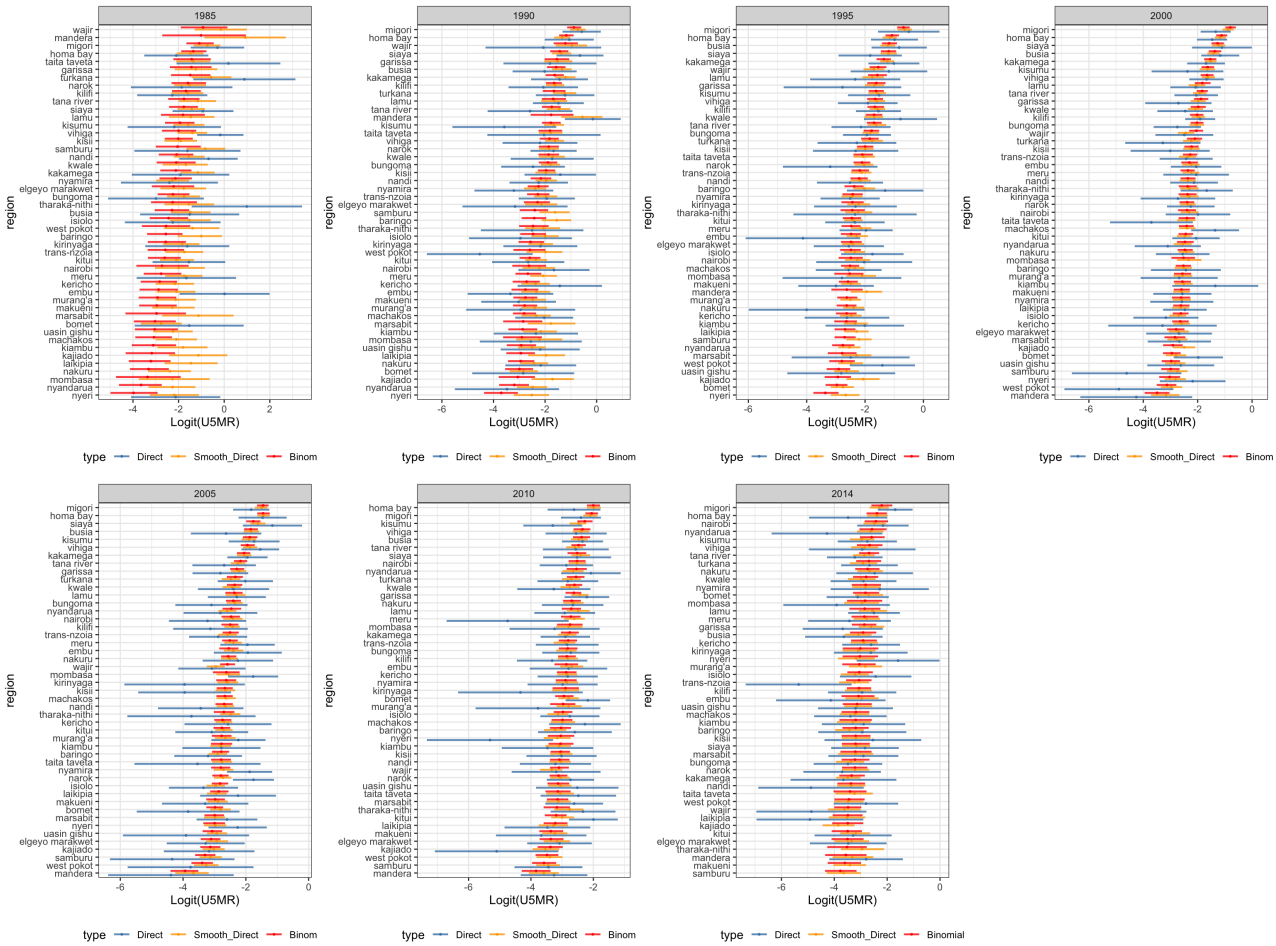
We plot the time series of U5MR for each county over time.

```
pos <- position_dodge(width = 0.2)
ggplot(results, aes(x = years, y = Est, color = type, ymin = lower, ymax = upper)) +
    geom_point(position = pos, alpha = 0.8, size = 0.5) + geom_errorbar(position = pos,
    size = 0.5, alpha = 0.8) + facet_wrap(~region, ncol = 7) + theme_bw() +
    theme(legend.position = "bottom") + scale_colour_manual(values = c("steelblue",
    "orange", "red"))
```



Here we compare selected years for all counties more closely, on the logit scale for easier comparison.

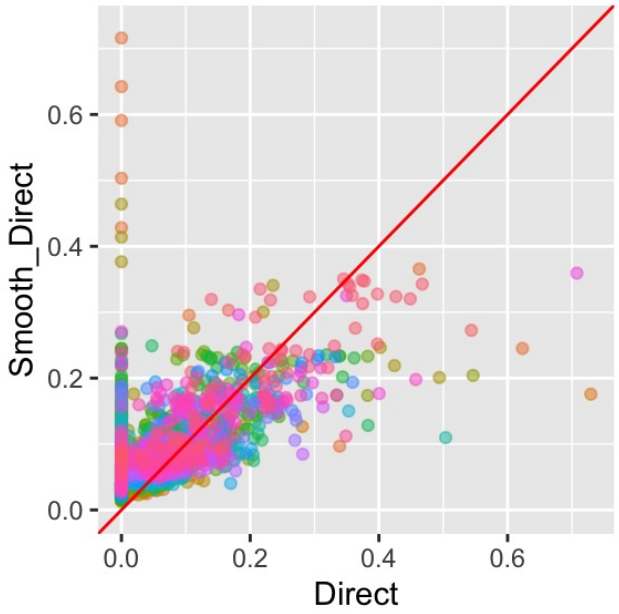
```
g <- NULL
for (i in 1:7) {
    t <- c(1985, 1990, 1995, 2000, 2005, 2010, 2014)[i]
    selectyear <- subset(results, type == "Binomial" & years == t)
    ordered <- selectyear[order(selectyear$Est), "region"]
    results$region <- factor(results$region, levels = ordered)
    pos <- position_dodge(width = 0.8)
    g[[i]] <- ggplot(subset(results, years == t), aes(y = logit(Est), x = region,
        color = type, ymin = logit(lower), ymax = logit(upper))) + geom_point(position = pos,
        size = 0.5) + geom_errorbar(position = pos, size = 0.8, width = 0, alpha = 0.8) +
        facet_grid(~years) + theme_bw() + theme(legend.position = "bottom") +
        ylab("Logit(U5MR)") + coord_flip() + scale_colour_manual(values = c("steelblue",
        "orange", "red"))
}
grid.arrange(grobs = g, ncol = 4)
```

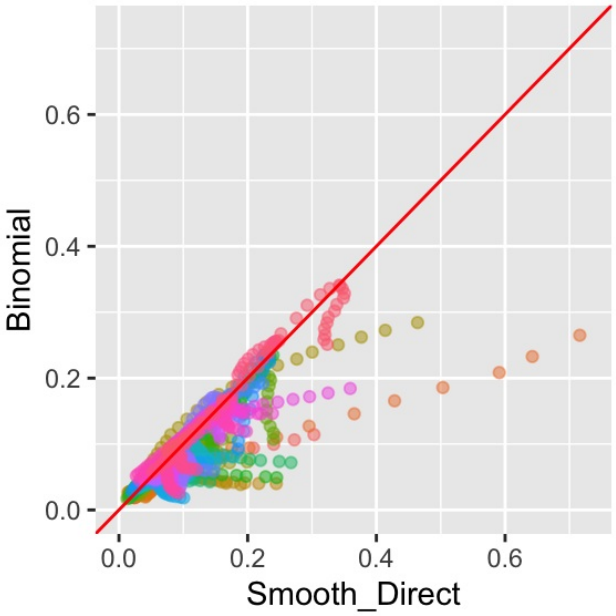


Finally, we combine all the points and compare the estimates together. Since some of the direct estimates do not exist, we replace the NA values to be 0 in those cases for the purpose of visualization (as can be seen in the left panel, where many dots lie on the vertical line where direct estimate is \(0\)).

```
library(tidyr)
range <- range(c(0, results$Est), na.rm = TRUE)
results.wide <- spread(results[, c(1, 2, 3, 6)], type, Est)
results.wide$Direct[is.na(results.wide$Direct)] <- 0

g1 <- ggplot(results.wide, aes(x = Direct, y = Smooth_Direct, color = region)) +
    geom_point(alpha = 0.5) + geom_abline(intercept = 0, slope = 1, color = "red") +
    xlim(range) + ylim(range) + theme(legend.position = "none")
g2 <- ggplot(results.wide, aes(x = Smooth_Direct, y = Binomial, color = region)) +
    geom_point(alpha = 0.5) + geom_abline(intercept = 0, slope = 1, color = "red") +
    xlim(range) + ylim(range) + theme(legend.position = "none")
grid.arrange(g1, g2, ncol = 2)
```


Smooth_Direct
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
Direct


Binomial
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
Smooth_Direct