Examples of Bayesian Small Area Estimation using Survey Data

Zehang Richard Li

2018/01/20

The `SUMMER` package offers a class of tools for small-area estimation with survey data in space and time. Such data are usually collected with complex stratified designs, which must be acknowledge in the analysis. In this vignette, we offer two main examples to illustrate spatial and spatial-temporal smoothing of design-based estimates:

- Naive, spatial, and spatial-temporal smoothing of design-based estimates using BRFSS data.
- Spatial-temporal smoothing under-5 child mortality rates (U5MR) using DHS example data.

The second example of estimating U5MR is the main purpose of this package, and involves more complex modeling steps. For many users, the first BRFSS example may be of more general interest, and provides an introduction to the idea of small area estimation with survey data. At the end of this vignette, we also provide a toy example of simulating spatially correlated data and performing spatial smoothing of design-based estimates.

Small Area Estimation with BRFSS Data

The Behavioral Risk Factor Surveillance System (BRFSS) is an annual telephone health survey conducted by the Centers for Disease Control and Prevention (CDC) that tracks health conditions and risk behaviors in the United States and its territories since 1984. The BRFSS sampling scheme is complex with high variability in the sampling weights. In this example, we estimate the prevalence of Type II diabetes in health reporting areas (HRAs) in King County, using BRFSS data. We will compare the weighted direct estimates, with simple spatial smoothed estimates (ignore weighting), and the smoothed and weighted estimates.

Load Package and 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.

```r
library(SUMMER)
if (!isTRUE(requireNamespace("INLA", quietly = TRUE))) {
    install.packages("INLA", repos = "https://www.math.ntnu.no/inla/R/stable")
}
data(BRFSS)
data(KingCounty)
```

`BRFSS` contains the full BRFSS dataset with $16,283$ observations. The `diab2` variable is the binary indicator of Type II diabetes, `strata` is the strata indicator and `rwt_llcp` is the final design weight. For the purpose of this analysis, we first remove records with missing HRA code or diabetes status from this dataset.

```r
BRFSS <- subset(BRFSS, !is.na(BRFSS$diab2))
BRFSS <- subset(BRFSS, !is.na(BRFSS$hracode))
```

`KingCounty` contains the map of the King County HRAs. In order to fit spatial smoothing model, we first need to compute the adjacency matrix for the HRAs, `mat`, and make sure both the column and row names correspond to the HRA names.

```r
library(spdep)
nb.r <- poly2nb(KingCounty, queen = F, row.names = KingCounty$HRA2010v2_)
mat <- nb2mat(nb.r, style = "B", zero.policy = TRUE)
colnames(mat) <- rownames(mat)
mat <- as.matrix(mat[1:dim(mat)[1], 1:dim(mat)[1]])
```

The Direct Estimates

Let y_i and m_i be the number of individuals flagged as having type II diabetes and the denominators in areas $i = 1, ..., n$. Ignoring the survey design, the naive estimates for the prevalence of Type II diabetes can be easily calculated as $\hat{p}_i = y_i/m_i$, with associated standard errors $\sqrt{\hat{p}_i(1-\hat{p}_i)/m_i}$. The design-based weighted estimates of p_i and the associated variances can be easily calculated using the `survey` package.

```r
library(survey)
design <- svydesign(ids = ~1, weights = ~rwt_llcp, strata = ~strata, data = BRFSS)
direct <- svyby(~diab2, ~hracode, design, svymean)
head(direct)
```

```
##                                                 hracode      diab2
## Auburn-North                               Auburn-North 0.10403154
## Auburn-South                               Auburn-South 0.23293289
## Ballard                                         Ballard 0.07047572
## Beacon/Gtown/S.Park                 Beacon/Gtown/S.Park 0.08083033
## Bear Creek/Carnation/Duvall Bear Creek/Carnation/Duvall 0.05166773
## Bellevue-Central                       Bellevue-Central 0.05914082
##                                     se
## Auburn-North                0.02147752
## Auburn-South                0.04897800
## Ballard                     0.02225241
## Beacon/Gtown/S.Park         0.02603522
## Bear Creek/Carnation/Duvall 0.01190146
## Bellevue-Central            0.01485885
```

The Smoothed Estimates

When the number of samples in each area is large, the design-based variance is usually small and the direct estimates will work well. However, when we have small sample from each area, we would like to perform some form of smoothing over the areas. For now, let us ignore the survey weights, we can consider the following Bayesian smoothing model:

$$
\begin{array}{rcl}
y_i|p_i & \sim & \text{Binomial}(m_i, p_i) \\
\theta_i & = & \log\left(\dfrac{p_i}{1-p_i}\right) = \mu + \epsilon_i + s_i, \\
\epsilon_i & \sim & N(0, \sigma_\epsilon^2) \\
s_i|s_j, j \in \text{ne}(i) & \sim & N\left(\bar{s_i}, \dfrac{\sigma_s^2}{n_i}\right).
\end{array}
$$

where n_i is the number of neighbors for area i, $\bar{s_i} = \frac{1}{n_i}\sum_{j \in \text{ne}(i)} s_j$, and hyperpriors are put on $\mu, \sigma_\epsilon^2, \sigma_s^2$. This simple smoothing model can be fitted using the `fitSpace()` function by specifying NULL for the survey parameters

```
smoothed <- fitSpace(data = BRFSS, geo = KingCounty, Amat = mat, family = "binomial",
    responseVar = "diab2", strataVar = NULL, weightVar = NULL, regionVar = "hracode",
    clusterVar = NULL, hyper = NULL, CI = 0.95)
```

The smoothed estimates of p_i and θ_i can be found in the `smooth` object returned by the function, and the direct estimates are stored in the `HT` object (without specifying survey weights, these are the simple binomial probabilities).

```
head(smoothed$smooth)
```

```
##                         region time      mean   variance    median
## 1                 Auburn-North   NA -1.853982 0.01841251 -1.852586
## 2                 Auburn-South   NA -1.453982 0.02697235 -1.453219
## 3                      Ballard   NA -2.671773 0.02108353 -2.669257
## 4           Beacon/Gtown/S.Park   NA -2.384320 0.02303002 -2.382882
## 5 Bear Creek/Carnation/Duvall   NA -2.566514 0.01724999 -2.563885
## 6             Bellevue-Central   NA -2.446837 0.02886851 -2.444425
##       lower     upper mean.original variance.original median.original
## 1 -2.123473 -1.590395    0.13618081      2.546956e-04      0.13550552
## 2 -1.777806 -1.132144    0.19087216      6.414494e-04      0.18961360
## 3 -2.964064 -2.394024    0.06524657      7.793095e-05      0.06484163
## 4 -2.688549 -2.090926    0.08517546      1.383684e-04      0.08454770
## 5 -2.830276 -2.315697    0.07190496      7.654875e-05      0.07156512
## 6 -2.787372 -2.120558    0.08053696      1.578492e-04      0.07983830
##   lower.original upper.original
## 1     0.10680742     0.16938411
## 2     0.14490034     0.24394345
## 3     0.04909929     0.08369460
## 4     0.06387088     0.11007297
## 5     0.05571725     0.09002422
## 6     0.05791285     0.10714353
```

```
head(smoothed$HT)
```

```
##      HT.est     HT.sd HT.variance   HT.prec HT.est.original
## 1 -1.812902 0.1726995  0.02982513 33.52878      0.14028777
## 2 -1.196804 0.1760789  0.03100377 32.25414      0.23204420
## 3 -2.639057 0.1701691  0.02895753 34.53333      0.06666667
## 4 -2.367124 0.2465033  0.06076389 16.45714      0.08571429
## 5 -2.600738 0.1681336  0.02826891 35.37455      0.06909091
## 6 -2.390291 0.2132685  0.04548346 21.98601      0.08391608
##   HT.variance.original   n  y                      region
## 1         0.0004338385 278 39                Auburn-North
## 2         0.0009845287 181 42                Auburn-South
## 3         0.0001121121 555 37                     Ballard
## 4         0.0003731778 210 18          Beacon/Gtown/S.Park
## 5         0.0001169406 550 38 Bear Creek/Carnation/Duvall
## 6         0.0002687908 286 24            Bellevue-Central
```

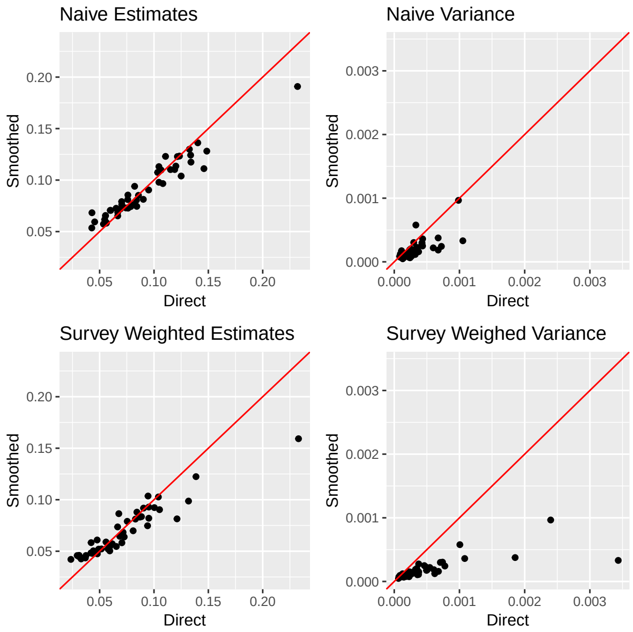
The Weighted and Smoothed Estimates

To account for the survey designs in the smoothing, we instead model the logit of the design-based direct estimates $\hat{p}_i \sim N(\theta_i, \hat{V}_i)$ directly, where both $\hat{p}_i$ and the asymptotic variance on the logit scale, $\hat{V}_i$, are assumed known. This model can be fit with

```
svysmoothed <- fitSpace(data = BRFSS, geo = KingCounty, Amat = mat, family = "binomial",
    responseVar = "diab2", strataVar = "strata", weightVar = "rwt_llcp",
    regionVar = "hracode", clusterVar = "~1", hyper = NULL, CI = 0.95)
```

Again, the design-based direct estimates and the smoothed estimates accounting for design can be obtained by svysmoothed$smooth and svysmoothed$HT. We can now compare the three types of estimates and their associated variance and it can be seen that smoothing reduces the variance of estimates significantly.

```
est <- data.frame(naive = smoothed$HT$HT.est.original,
                  weighted = svysmoothed$HT$HT.est.original,
                  smooth = smoothed$smooth$mean.original,
                  weightedsmooth = svysmoothed$smooth$mean.original)
var <- data.frame(naive = smoothed$HT$HT.variance.original,
                  weighted = svysmoothed$HT$HT.variance.original,
                  smooth = smoothed$smooth$variance.original,
                  weightedsmooth = svysmoothed$smooth$variance.original)
l1 <- range(est)
l2 <- range(var)
g1 <- ggplot(est, aes(x = naive, y = smooth)) + geom_point() +
        geom_abline(slope = 1, intercept = 0, color = "red") +
        ggtitle("Naive Estimates") + xlab("Direct") + ylab("Smoothed")+  xlim(l1) + ylim(l1)
g2 <- ggplot(var, aes(x = naive, y = weightedsmooth)) + geom_point() +
       geom_abline(slope = 1, intercept = 0, color = "red") +
       ggtitle("Naive Variance") + xlab("Direct") + ylab("Smoothed") + xlim(l2) + ylim(l2)
g3 <- ggplot(est, aes(x = weighted, y = weightedsmooth)) + geom_point() +
        geom_abline(slope = 1, intercept = 0, color = "red") +
        ggtitle("Survey Weighted Estimates") + xlab("Direct") + ylab("Smoothed") + xlim(l1) + ylim(l1)
g4 <- ggplot(var, aes(x = weighted, y = weightedsmooth)) + geom_point() +
        geom_abline(slope = 1, intercept = 0, color = "red") +
        ggtitle("Survey Weighed Variance") + xlab("Direct") + ylab("Smoothed") + xlim(l2) + ylim(l2)
library(gridExtra)
grid.arrange(grobs = list(g1, g2, g3, g4), ncol = 2)
```



Customized prior distribution

The `fitSpace` function has some default hyperprior choices built in. For Binomial models, we use $Ga(0.5, 0.001488)$ on the latent precisions, which leads to a 95% prior interval for the residual odds ratio between $[0.5, 2]$, and for Gaussian models we use the default $Ga(1, 5E-5)$ prior from INLA. There are two ways to customize this default hyperprior choice. To simply update the hyper parameters of the Gamma prior, we can simply use the `hyper.besag` and `hyper.iid` arguments. For example,

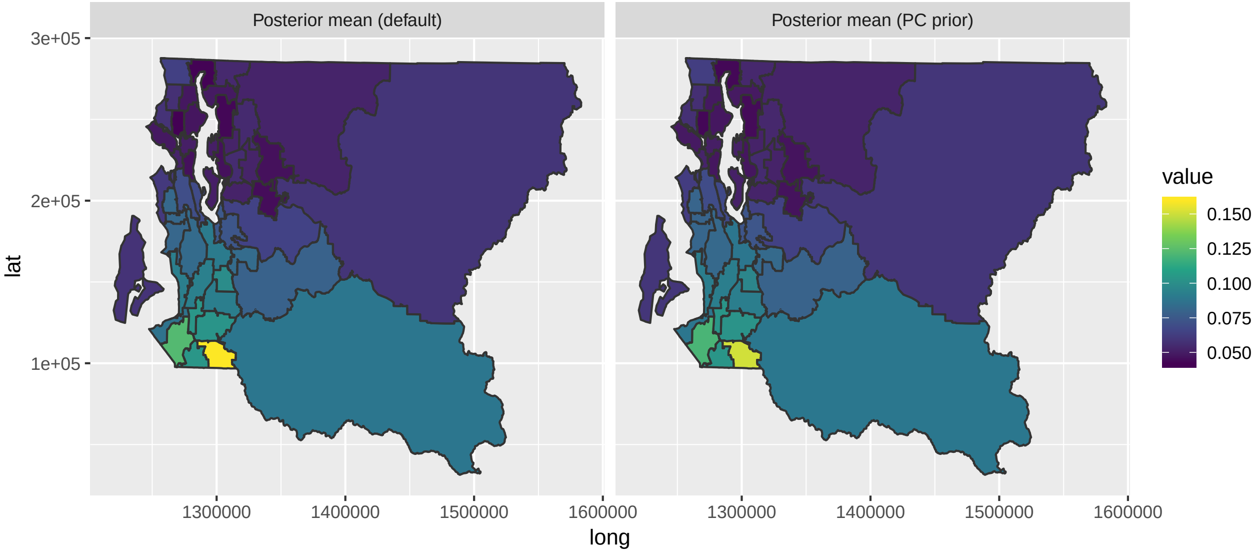
```
svysmooth.1 <- fitSpace(data = BRFSS, geo = KingCounty, Amat = mat, family = "binomial",
    responseVar = "diab2", strataVar = "strata", weightVar = "rwt_llcp",
    regionVar = "hracode", clusterVar = "~1", hyper = NULL, CI = 0.95,
    hyper.besag = c(0.5, 0.01), hyper.iid = c(0.5, 0.01))
```

Or we can change the random effect model entirely using the `newformula` argument. To use this option, we will need to use the internal indicator for region, `rregion.struct` as the index. This index variable name can also be observed from the fitted object by `summary(svysmoothed$fit)`. For example, we can reassign a different parameterization of the BYM model. For users familiar with INLA syntax, this allows more possibilities of expanding the modeling framework.

```
 newformula <- "f(region.struct, model = 'bym2', graph = Amat,
               constr = TRUE,scale.model = TRUE, hyper = list(
               phi = list(prior = 'pc', param = c(0.5 , 2/3)
                   , initial = -3),
               prec = list(prior = 'pc.prec', param = c(0.2/0.31 , 0.01)
                   , initial = 5)))"
 svysmooth.2 <- fitSpace(data = BRFSS, geo = KingCounty, Amat = mat,
  family = "binomial", responseVar="diab2",
  strataVar="strata", weightVar="rwt_llcp", regionVar="hracode",
  clusterVar = "~1", hyper=NULL,  CI = 0.95,
  newformula = newformula)
```

Another way to visualize one or more metrics on the map is by the `mapPlot()` function, for example,

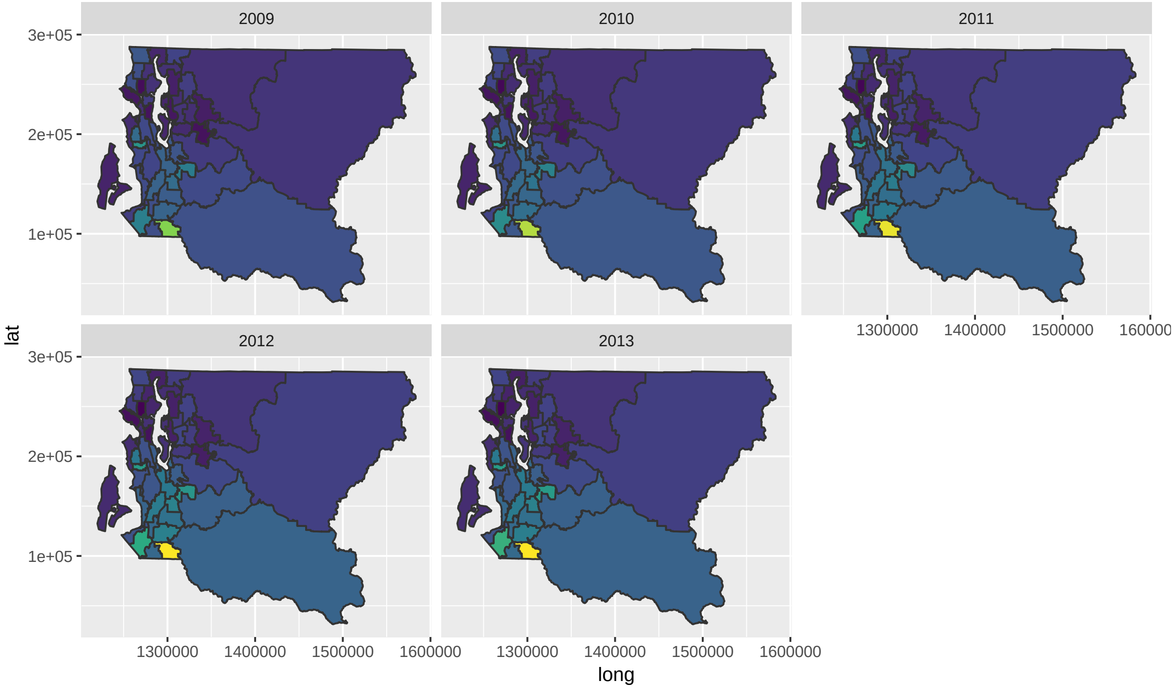
```
toplot <- svysmoothed$smooth
toplot$newprior <- svysmooth.2$smooth$mean.original
mapPlot(data = toplot, geo = KingCounty, variables = c("mean.original",
    "newprior"), labels = c("Posterior mean (default)", "Posterior mean (PC prior)"),
    by.data = "region", by.geo = "HRA2010v2_")
```



Small Area Estimation in Space and Time

When data consist of observations from different time periods, we can extend the framework to smooth estimates over both space and time. The space-time interaction terms are modeled by the type I-IV interactions.

```
svysmoothed.year <- fitSpace(data = BRFSS, geo = KingCounty, Amat = mat,
    family = "binomial", responseVar = "diab2", strataVar = "strata", weightVar = "rwt_llcp",
    regionVar = "hracode", clusterVar = "~1", timeVar = "year", time.model = "rw1",
    type.st = 4)
mapPlot(data = svysmoothed.year$smooth, geo = KingCounty, values = "mean.original",
    variables = "time", by.data = "region", by.geo = "HRA2010v2_", is.long = TRUE)
```



Similarly the default priors can also be extended using the `newformula` argument.

U5MR Estimation in Space and Time

Load Data

`DemoData` contains model survey data provided by DHS. Note that this data is fake, and does not represent any real country's data. Data similar to the `DemoData` data used in this vignette can be obtained by using `getBirths`. `DemoMap` contains geographic data from the 1995 Uganda Admin 1 regions defined by DHS. Data similar to the `DemoMap` data used in this vignette can be obtained by using `read_shape`.

```
data(DemoData)
data(DemoMap)
geo <- DemoMap$geo
mat <- DemoMap$Amat
```

`DemoData` is a list of 5 data frames where each row represent one person-month record and contains the 8 variables as shown below. Notice that `time` variable is turned into 5-year bins from `80-84` to `10-14`.

```
summary(DemoData)
```

```
##      Length Class      Mode
## 1999 8      data.frame list
## 2003 8      data.frame list
## 2007 8      data.frame list
## 2011 8      data.frame list
## 2015 8      data.frame list
```

```
head(DemoData[[1]])
```

```
##   clustid id  region  time  age  weights        strata died
## 1       1  1 eastern 00-04    0 1.057703 eastern.rural    0
## 2       1  1 eastern 00-04 1-11 1.057703 eastern.rural    0
## 3       1  1 eastern 00-04 1-11 1.057703 eastern.rural    0
## 4       1  1 eastern 00-04 1-11 1.057703 eastern.rural    0
## 5       1  1 eastern 00-04 1-11 1.057703 eastern.rural    0
## 6       1  1 eastern 00-04 1-11 1.057703 eastern.rural    0
```

DemoData is obtained by processing the raw DHS birth data (in .dta format) in R. The raw file of birth recodes can be downloaded from the DHS website https://dhsprogram.com/data/Download-Model-Datasets.cfm. For this example dataset, no registration is needed. For real DHS survey datasets, permission to access needs to be registered with DHS directly. DemoData contains a small sample of the observations in this dataset randomly assigned to 5 example DHS surveys.

Here we demonstrate how to split the raw data into person-month format from. Notice that to read the file from early version of stata, the package readstata13 is required. The following script is based on the example dataset ZZBR62FL.DTA available from the DHS website. We use the interaction of v024 and v025 as the strata indicator for the purpose of demonstration.

```
library(readstata13)
my_fp <- "data/ZZBR62DT/ZZBR62FL.DTA"
dat <- getBirths(filepath = my_fp, surveyyear = 2015, strata = c("v024",
    "v025"))
dat <- dat[, c("v001", "v002", "v024", "per5", "ageGrpD", "v005", "strata",
    "died")]
colnames(dat) <- c("clustid", "id", "region", "time", "age", "weights",
    "strata", "died")
```

Horvitz-Thompson estimators of U5MR

Next, we obtain Horvitz-Thompson estimators using countrySummary_mult.

```
years <- levels(DemoData[[1]]$time)

data_multi <- countrySummary_mult(births = DemoData, years = years, idVar = "id",
    regionVar = "region", timeVar = "time", clusterVar = "~clustid+id",
    ageVar = "age", weightsVar = "weights", geo.recode = NULL)
```

Before fitting the model, we also aggregate estimates based on different surveys into a single set of estimates, using the inverse design-based variances as the weights.

```
dim(data_multi)
```

```
## [1] 150  11
```

```r
data <- aggregateSurvey(data_multi)
dim(data)
```

```
## [1] 30 10
```

National estimates of U5MR

Using our adjacency matrix, we first simulate hyperpriors using `simhyper`. The default INLA analysis scales the marginal variance of all structured random effects, so we only need to one set of hyperparameters with `only.iid` set to true.

```r
priors <- simhyper(R = 2, nsamp = 1e+05, nsamp.check = 5000, Amat = mat,
    only.iid = TRUE)
```

Now we are ready to fit the models. First, we ignore the subnational estimates, and fit a model with temporal random effects only. In this part, we use the subset of data region variable being "All".

In fitting this model, we first define the list of time periods we wish to project the estimates on. First we can fit a Random Walk 2 only model defined on the 5-year period.

```r
years.all <- c(years, "15-19")
fit1 <- fitINLA(data = data, geo = NULL, Amat = NULL, year_names = years.all,
    year_range = c(1985, 2019), priors = priors, rw = 2, is.yearly = FALSE,
    m = 5)
```

We can also estimate the Random Walk 2 random effects on the yearly scale.

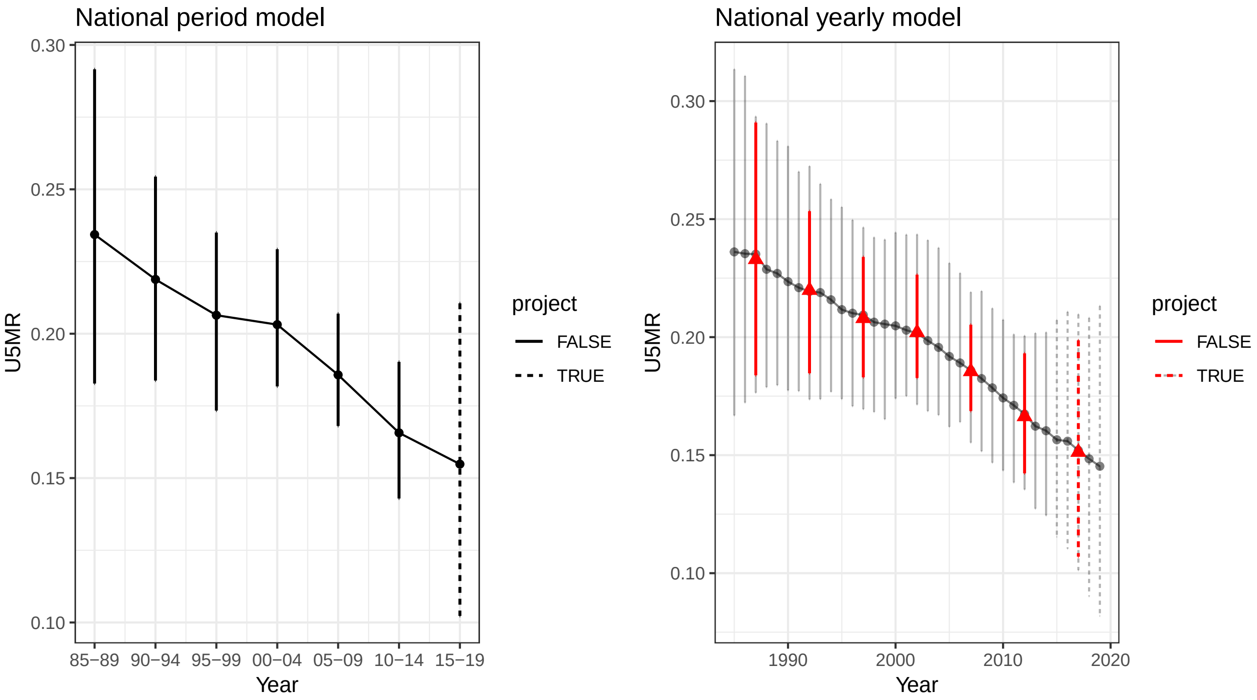
```r
fit2 <- fitINLA(data = data, geo = NULL, Amat = NULL, year_names = years.all,
    year_range = c(1985, 2019), priors = priors, rw = 2, is.yearly = TRUE,
    m = 5)
```

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

```r
out1 <- projINLA(fit1, is.yearly = FALSE)
out2 <- projINLA(fit2, is.yearly = TRUE)
```

We can compare the results visually using the function below.

```r
library(ggplot2)
library(gridExtra)
g <- NULL
g[[1]] <- plot(out1, is.yearly = FALSE, is.subnational = FALSE) + ggtitle("National period model")
g[[2]] <- plot(out2, is.yearly = TRUE, is.subnational = FALSE) + ggtitle("National yearly model")
grid.arrange(grobs = g, ncol = 2)
```



Subnational estimates of U5MR

Similarly we can fit the full model on all subnational regions as well. First, we fit the Random Walk 2 model defined on the 5-year period.

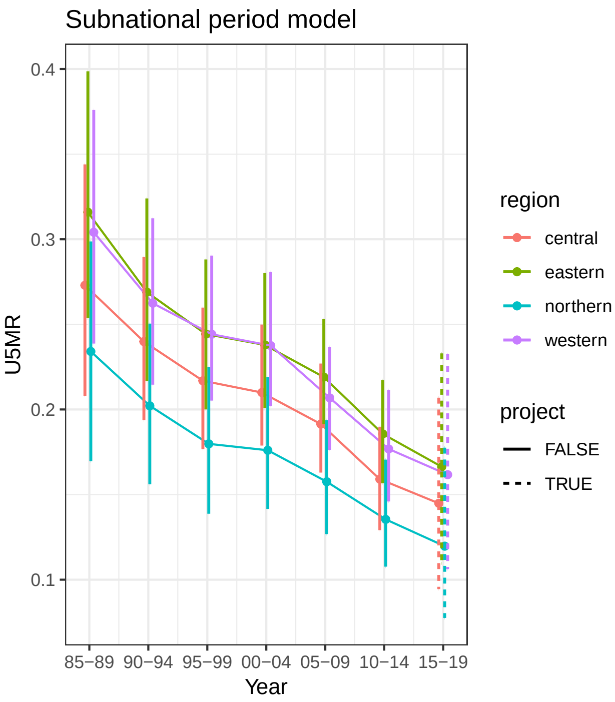
```
fit3 <- fitINLA(data = data, geo = geo, Amat = mat, year_names = years.all,
    year_range = c(1985, 2019), priors = priors, rw = 2, is.yearly = FALSE)
out3 <- projINLA(fit3, Amat = mat, is.yearly = FALSE)
```

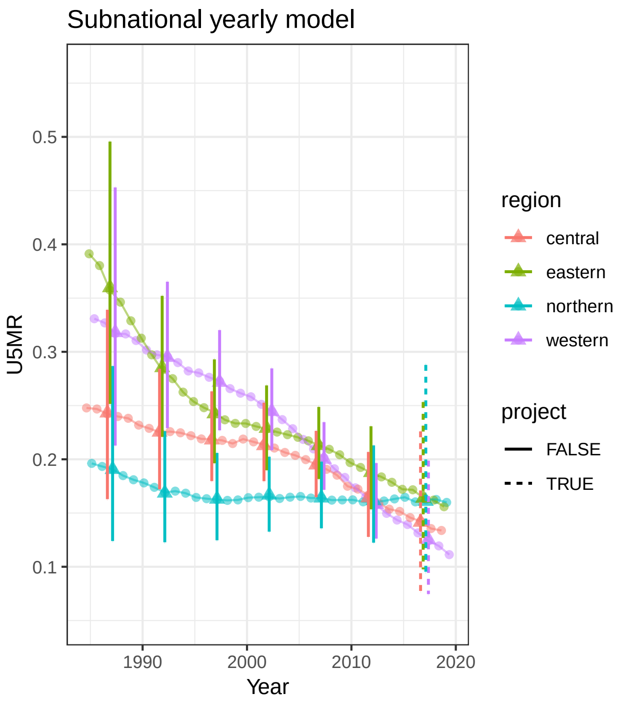
Similarly we can also estimate the Random Walk 2 random effects on the yearly scale.

```
fit4 <- fitINLA(data = data, geo = geo, Amat = mat, year_names = years.all,
    year_range = c(1985, 2019), priors = priors, rw = 2, is.yearly = TRUE,
    m = 5, type.st = 4)
out4 <- projINLA(fit4, Amat = mat, is.yearly = TRUE)
```

The figures below shows the comparison of the subnational model with different temporal scales.

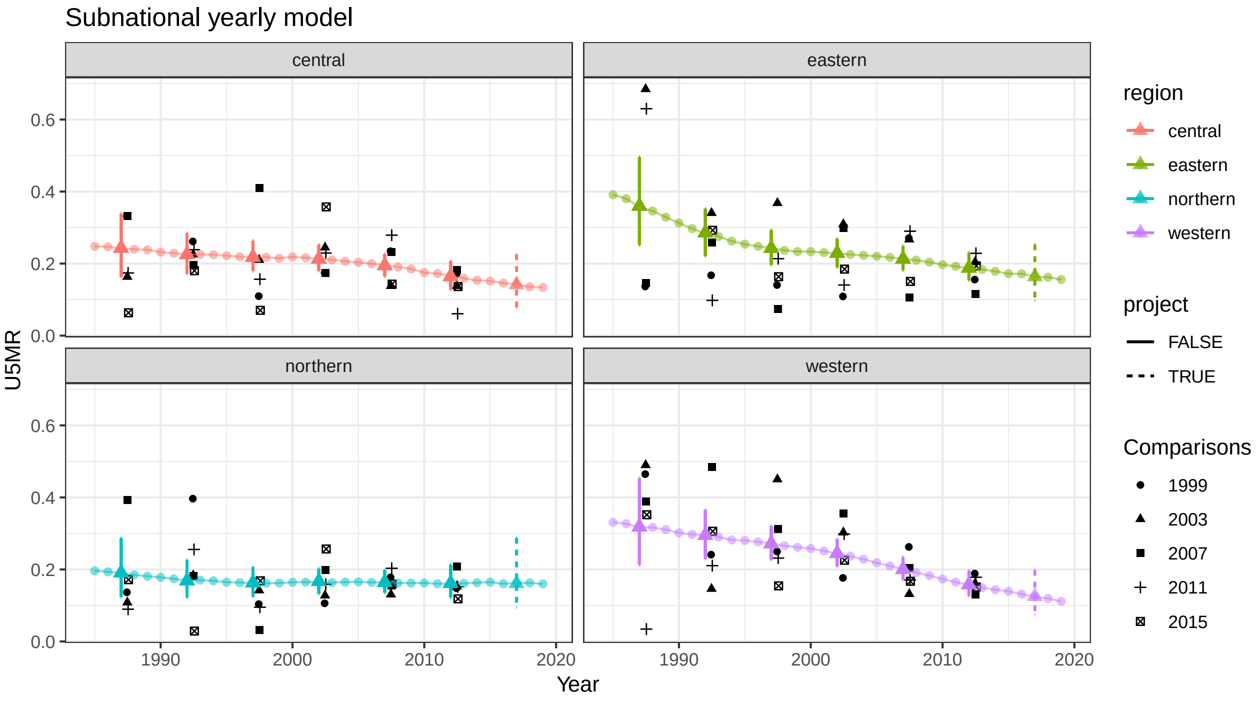
```
g2 <- NULL
g2[[1]] <- plot(out3, is.yearly = FALSE, is.subnational = TRUE) + ggtitle("Subnational period model")
g2[[2]] <- plot(out4, is.yearly = TRUE, is.subnational = TRUE) + ggtitle("Subnational yearly model")
grid.arrange(grobs = g2, ncol = 2)
```





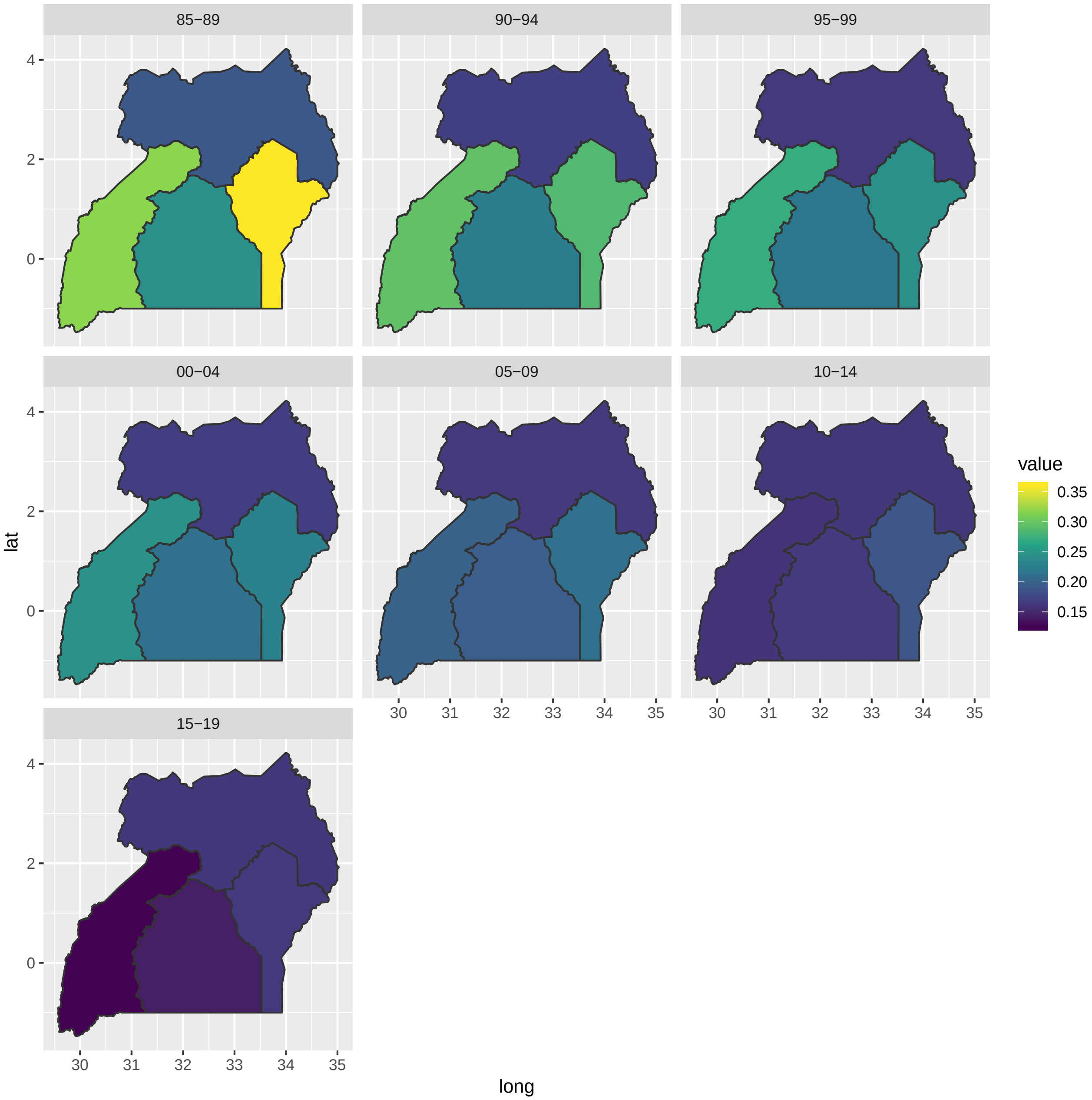
We can also add back the direct estimates for comparison.

```
plot(out4, is.yearly = TRUE, is.subnational = TRUE, data.add = data_multi,
    option.add = list(point = "u5m", by = "surveyYears")) + ggplot2::ggtitle("Subnational yearly model"
    facet_wrap(~region)
```



Finally, we show the estimates over time on maps.

```
mapPlot(data = subset(out4, is.yearly == F), geo = DemoMap$geo, variables = c("years"),
    values = c("med"), by.data = "region", by.geo = "NAME_final", is.long = TRUE)
```



Simulate spatial(temporal) random effects

In this section we simulate spatially correlated data and perform spatial smoothing. The simulation scheme in this section can be extended to temporal and spatial-temporal case as well.

As an illustration, we use a simulated dataset created from the model survey data, and a Kenya Admin 1 map with 8 regions.

```r
data(DemoData2)
data(DemoMap2)
regions <- colnames(DemoMap2$Amat)
```

We first generate some additional synthetic normally distributed variable for height for each observation. Suppose we denote the height of observation k in area i to be x_{ik}, and the associated design weight to be w_{ik}.

Under the design-based approach to inference, we can calculate the weighted estimator of mean height to be

$$\hat{\mu_i} = \frac{\sum_k w_{ik} x_{ik}}{\sum_k w_{ik}}$$

and the associated variance $\widehat{var}(\hat{\mu_i})$. We then use INLA to fit the following Bayesian hierarchical model:

$$\begin{aligned} \hat{\mu_i} &\sim \text{Normal}(\mu_i, \widehat{var}(\hat{\mu_i})) \\ \mu_i &= \beta + \epsilon_i + \delta_i, \\ \epsilon_i &\sim \text{Normal}(0, \sigma^2_\epsilon) \\ \delta_i &\sim \text{ICAR}(\sigma^2_\delta) \end{aligned}$$

To simulate from this generative model, we first simulate the mean height for each region from the ICAR random fields as follows

```
u <- rst(n = 1, type = "s", Amat = DemoMap2$Amat)
mu <- 5 + u
```

We generate data by

```
DemoData2$height <- rnorm(dim(DemoData2)[1]) * 8 + mu[match(DemoData2$region,
    regions)]
```

We can use the `fitspace()` function to obtain both the survey-weighted direct estimates and the smoothed estimates from INLA.

```
fit <- fitSpace(data = DemoData2, geo = DemoMap2$geo, Amat = DemoMap2$Amat,
    family = "gaussian", responseVar = "height", strataVar = "strata",
    weightVar = "weights", regionVar = "region", clusterVar = "~clustid+id",
    hyper = NULL, CI = 0.95)
```

The direct estimates of the average height, i.e., $\hat{\mu_i}$ accounting for survey design are

```
fit$HT[, c("HT.est", "HT.variance", "region")]
```

```
##     HT.est HT.variance       region
## 2 5.028752  0.09207772      nairobi
## 1 5.356792  0.05140550      central
## 4 5.633079  0.05577057        coast
## 3 3.192389  0.13129811      eastern
## 6 5.472257  0.13709734       nyanza
## 8 6.389748  0.08934184  rift valley
## 7 3.432467  0.12759646      western
## 5 4.807714  0.06717659 northeastern
```

The smoothed estimates of the average height, i.e., $\hat{\mu_i}$ accounting for survey design are

```
fit$smooth[, c("mean", "variance", "region")]
```

```
##       mean   variance       region
## 1 5.020195 0.08443119      nairobi
## 2 5.333302 0.04904822      central
## 3 5.590945 0.05331936        coast
## 4 3.419469 0.12873958      eastern
## 5 5.400409 0.12216072       nyanza
## 6 6.254055 0.08670370  rift valley
## 7 3.621649 0.12237214      western
## 8 4.816855 0.06304492 northeastern
```