Space-time Smoothing in R

Spatial smoothing: the generic case

Space-time smoothing: NMR

Space-time smoothing: U5MR

Bayesian Subnational Estimation using Complex Survey Data

Space-time Smoothing in R

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 two simulated datasets to illustrate three different scenarios of small area estimation (SAE):

- Spatial smoothing of the prevalence of a binary indicator.
- Space-time smoothing of neonatal mortality rates (NMR).
- Space-time smoothing of under-5 mortality rates (U5MR).

Spatial smoothing: the generic case

In the first example, we will use the simulated dataset `DemoData2` in the SUMMER package.

```
library(SUMMER)
library(sp)
library(ggplot2)
library(gridExtra)
data(DemoData2)
head(DemoData2)
```

```
##   clustid id  region age weights        strata tobacco.use
## 1       1  1 nairobi  30     1.1 nairobi.urban           0
## 2       1  3 nairobi  22     1.1 nairobi.urban           0
## 3       1  4 nairobi  42     1.1 nairobi.urban           0
## 4       2  4  nyanza  25     1.1  nyanza.urban           0
## 5       1  5 nairobi  25     1.1 nairobi.urban           0
## 6       1  6 nairobi  37     1.1 nairobi.urban           0
```

`DemoData2` simulates a survey from a stratified cluster sampling design similar to the DHS surveys. Suppose we are interested in the prevalence of tobacco usage.

Naive estimates

Let \(y_i\) and \(n_i\) denote the number of individuals using tobacco and the total number of people in areas i = 1, …, n, respectively. Ignoring the survey design, the naive estimate for the prevalence of tobacco usage is \[\hat{p}_i = y_i / n_i, \] with associated variance being \(p_i(1 − p_i)/n_i\). This naive estimate can be easily calculated by tabulating the data.

```
naive <- aggregate(tobacco.use ~ region, data = DemoData2, FUN = mean)
```

Smoothed (naive) estimates

Ignoring survey design for now, we may fit the following simple smoothing model for a simple random sample:

\[\begin{align} y_i | p_i &\sim \mbox{Binomial}(n_i, p_i), \\ \mbox{logit}(p_i) &= \mu + s_i + \epsilon_i, \end{align}\]

where the \(s_i\) and \(\epsilon_i\) are assumed to be following the ICAR and independent priors respectively. That is, the logit of prevalence can be parameterized by the BYM model.

We can fit this model with the `fitGeneric` function in the SUMMER package. Notice that in order to perform the smoothing, we need to construct a spatial adjacency matrix of the regions. The associated map with this simulated dataset is the province map of Kenya, as included in the package as well.

```
data(DemoMap2)
smoothed <- fitGeneric(data = DemoData2, geo = DemoMap2$geo, Amat = DemoMap2$Amat,
    responseType = "binary", responseVar = "tobacco.use", regionVar = "region",
    strataVar = NULL, weightVar = NULL, clusterVar = NULL, CI = 0.95)
```

For this model, the smoothed estimates are saved in `smoothed$smooth`, and the naive estimates are saved in `smoothed$HT`.

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

```
##        region time mean variance median lower upper mean.original
## 1     nairobi   NA -3.5    0.023   -3.5  -3.8  -3.2         0.031
## 2     central   NA -3.2    0.012   -3.2  -3.4  -3.0         0.040
## 3       coast   NA -2.6    0.017   -2.6  -2.9  -2.4         0.067
## 4     eastern   NA -3.2    0.024   -3.2  -3.5  -2.9         0.040
## 5      nyanza   NA -3.4    0.030   -3.4  -3.7  -3.1         0.033
## 6 rift valley   NA -2.9    0.026   -2.9  -3.2  -2.5         0.055
##   variance.original median.original lower.original upper.original
## 1          0.000020           0.031          0.023          0.040
## 2          0.000017           0.039          0.032          0.048
## 3          0.000065           0.067          0.052          0.083
## 4          0.000034           0.039          0.029          0.051
## 5          0.000031           0.033          0.023          0.044
## 6          0.000069           0.054          0.040          0.072
```

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

```
##   HT.est HT.sd HT.variance HT.prec HT.est.original HT.variance.original
## 2   -3.5  0.17       0.029      35           0.028             0.000021
## 1   -3.2  0.12       0.014      72           0.039             0.000020
## 4   -2.5  0.12       0.015      65           0.073             0.000070
## 3   -3.2  0.18       0.032      31           0.038             0.000043
## 6   -3.4  0.20       0.040      25           0.031             0.000036
## 8   -2.7  0.17       0.027      37           0.061             0.000090
##      n  y      region
## 2 1287 36     nairobi
## 1 1915 75     central
## 4  964 70       coast
## 3  843 32     eastern
## 6  839 26      nyanza
## 8  639 39 rift valley
```

Weighted estimates

Both the naive and smoothed estimates described before fail to account for the survey design and are thus subject to bias and incorrect variance estimation. To account for the different design weights associated with each sample, we can estimate the survey weighted estimates (the Horvitz-Thompson estimates) using the survey package.

```
library(survey)
design <- svydesign(ids = ~clustid + id, weights = ~weights, strata = ~strata,
    data = DemoData2)
direct <- svyby(~tobacco.use, ~region, design, svymean)
head(direct)
```

```
##                    region tobacco.use     se
## central           central       0.043 0.0079
## nairobi           nairobi       0.027 0.0057
## eastern           eastern       0.035 0.0114
## coast               coast       0.074 0.0088
## northeastern northeastern       0.037 0.0061
## nyanza             nyanza       0.028 0.0067
```

Smoothed and weighted estimates

Finally, we can smooth (the logit of) the weighted estimates directly, and obtain smoothed estimates that account for survey designs. That is, we fit the following model,

\[\begin{align} \mbox{logit}(\hat{p_i}) &\sim \mbox{Normal}(\theta_i, \hat{V}_i), \\ \theta_i &= \mu + s_i + \epsilon_i, \end{align}\] where \(\hat{p_i}\) and \(\hat{V}_i\) are the design-based weighted estimate of the prevalence at region \(i\), and the associated variance of its logit, respectively. Using the `fitGeneric` function, we now need to specify survey designs in order to fit this model

```
smoothweighted <- fitGeneric(data = DemoData2, geo = DemoMap2$geo, Amat = DemoMap2$Amat,
    responseType = "binary", responseVar = "tobacco.use", regionVar = "region",
    strataVar = "strata", weightVar = "weights", clusterVar = "~clustid+id",
    CI = 0.95)
```

We can again obtain the Horvitz-Thompson estimates and the spatially smoothed estimates using

```
head(smoothweighted$HT)
```

```
##   HT.est HT.sd HT.variance HT.prec HT.est.original HT.variance.original  n
## 2   -3.6  0.21       0.046    21.9           0.027             0.000033 NA
## 1   -3.1  0.19       0.037    26.8           0.043             0.000062 NA
## 4   -2.5  0.13       0.017    59.9           0.074             0.000078 NA
## 3   -3.3  0.34       0.116     8.6           0.035             0.000129 NA
## 6   -3.5  0.24       0.060    16.7           0.028             0.000045 NA
## 8   -2.6  0.22       0.049    20.4           0.072             0.000217 NA
##    y      region
## 2 NA     nairobi
## 1 NA     central
## 4 NA       coast
## 3 NA     eastern
## 6 NA      nyanza
## 8 NA rift valley
```

```
head(smoothweighted$smooth)
```

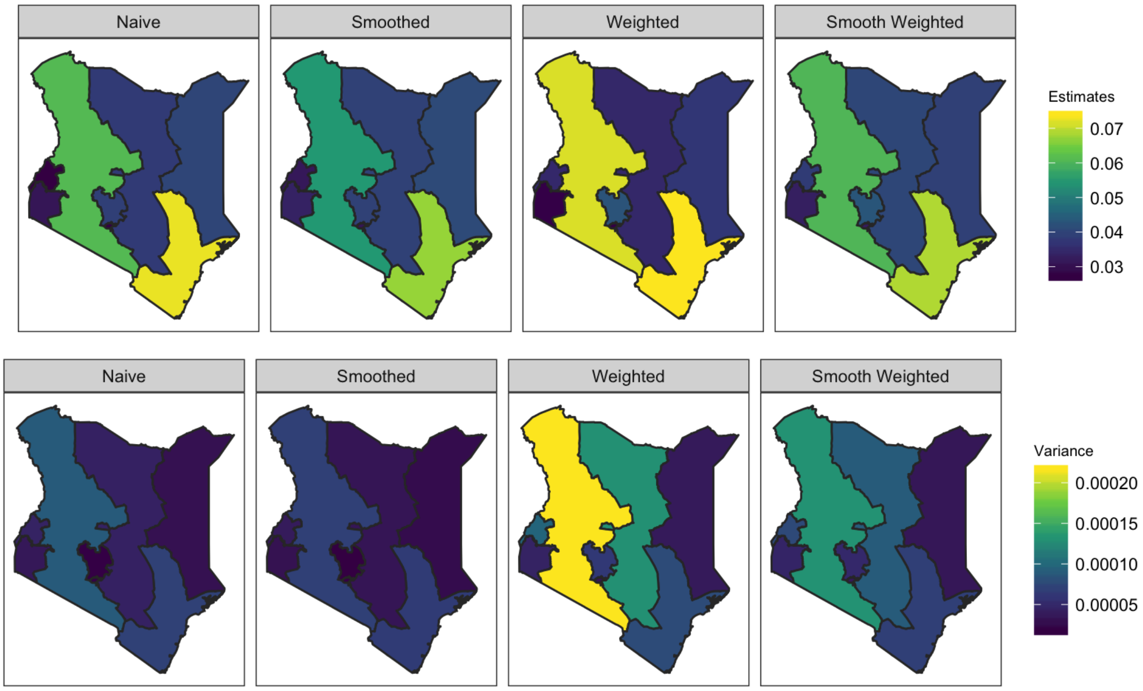
```
##        region time mean variance median lower upper mean.original
## 1     nairobi   NA -3.4    0.038   -3.4  -3.8  -3.1         0.032
## 2     central   NA -3.1    0.029   -3.1  -3.4  -2.8         0.043
## 3       coast   NA -2.6    0.017   -2.6  -2.9  -2.4         0.069
## 4     eastern   NA -3.2    0.061   -3.2  -3.7  -2.7         0.040
## 5      nyanza   NA -3.4    0.047   -3.4  -3.8  -3.0         0.033
## 6 rift valley   NA -2.8    0.039   -2.8  -3.1  -2.4         0.060
##   variance.original median.original lower.original upper.original
## 1          0.000036           0.032          0.022          0.045
## 2          0.000050           0.043          0.031          0.059
## 3          0.000068           0.068          0.054          0.086
## 4          0.000090           0.039          0.024          0.061
## 5          0.000047           0.032          0.021          0.048
## 6          0.000131           0.059          0.042          0.086
```

We now combine the four estimates and compare the results. First we construct a new data frame that hosts all four set of estimates and their variances.

```
prev <- NULL
prev <- rbind(prev, data.frame(region = smoothed$HT$region,
                               mean = smoothed$HT$HT.est.original,
                               var = smoothed$HT$HT.variance.original,
                               type = "Naive"))
prev <- rbind(prev, data.frame(region = smoothed$smooth$region,
                               mean = smoothed$smooth$mean.original,
                               var = smoothed$smooth$variance.original,
                               type = "Smoothed"))
prev <- rbind(prev, data.frame(region = smoothweighted$HT$region,
                               mean = smoothweighted$HT$HT.est.original,
                               var = smoothweighted$HT$HT.variance.original,
                               type = "Weighted"))
prev <- rbind(prev, data.frame(region = smoothweighted$smooth$region,
                               mean = smoothweighted$smooth$mean.original,
                               var = smoothweighted$smooth$variance.original,
                               type = "Smooth Weighted"))
```

We plot the estimates and variances on the map using function `mapPlot`. The spatial smoothing and reduced variance can be easily seen from the following plots.

```
g1 <- mapPlot(prev, geo = DemoMap2$geo, by.data = "region", by.geo = "REGNAME",
    variables = "type", values = "mean", is.long = TRUE, legend.label = "Estimates",
    ncol = 4)
g2 <- mapPlot(prev, geo = DemoMap2$geo, by.data = "region", by.geo = "REGNAME",
    variables = "type", values = "var", is.long = TRUE, legend.label = "Variance",
    ncol = 4)
grid.arrange(g1, g2, ncol = 1)
```



Space-time smoothing: NMR

To extend the spatial smoothing to the case of space-time smoothing, we will look at another simulated example of estimating the neonatal mortality rates. The dataset `DemoData` in the SUMMER package consist of simulated full birth history of five surveys. As an illustration of space-time smoothing of the prevalence of a binary indicator, we will look at only the first survey.

```
data(DemoData)
data <- DemoData[[1]]
head(data)
```

```
##   clustid id  region  time  age weights        strata died
## 1       1  1 eastern 00-04    0     1.1 eastern.rural    0
## 2       1  1 eastern 00-04 1-11     1.1 eastern.rural    0
## 3       1  1 eastern 00-04 1-11     1.1 eastern.rural    0
## 4       1  1 eastern 00-04 1-11     1.1 eastern.rural    0
## 5       1  1 eastern 00-04 1-11     1.1 eastern.rural    0
## 6       1  1 eastern 00-04 1-11     1.1 eastern.rural    0
```

The data has been arranged into the person-month format where each row represent one person-month record and contains the \(8\) variables as shown below. To calculate the neonatal mortality rates, we only need to know whether an child survived the first month. So we will subset our data to contain only the first age group.

```
data <- subset(data, age == 0)
```

We will again use the function `fitGeneric` to obtain the design-based and smoothed estimates of NMR. We will add the temporal component using the `timeVar` argument and choose the random effect for the temporal model (RW2), and the space-time interaction model (Type IV interaction). Notice that `time` variable is turned into 5-year bins from `80-84` to `10-14`. We need to first create a time index column first.

```
data$time.id <- match(data$time, levels(data$time))
fit <- fitGeneric(data = data, geo = DemoMap$geo, Amat = DemoMap$Amat, responseType = "binary",
    responseVar = "died", regionVar = "region", strataVar = "strata", weightVar = "weights",
    clusterVar = "~clustid+id", CI = 0.95, timeVar = "time.id", time.model = "rw1",
    type.st = 4)
```

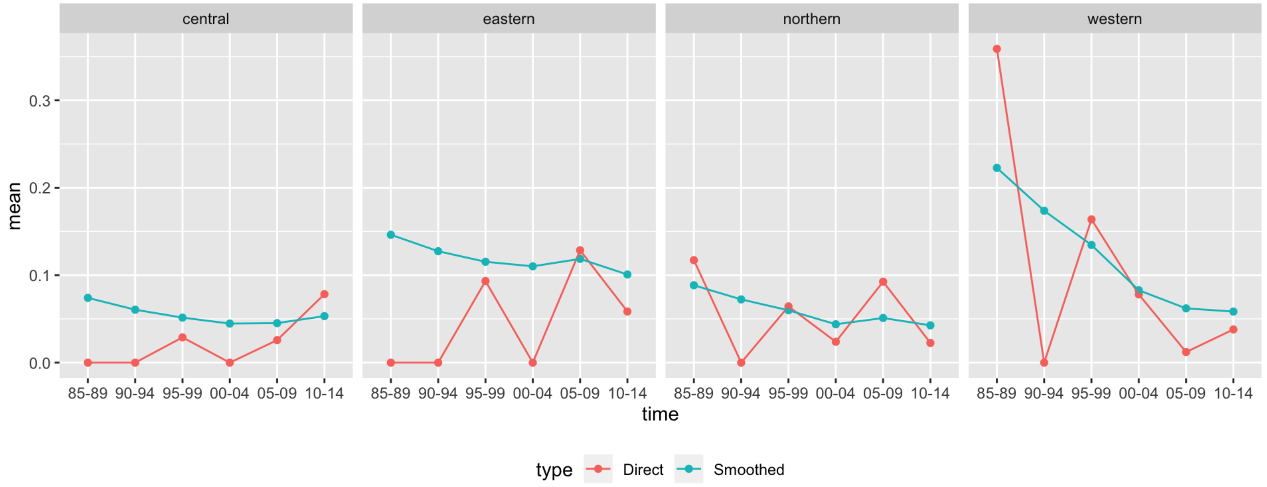
We then combine the direct and smoothed estimates into the data frame `nmr`.

```
nmr <- NULL
nmr <- rbind(nmr, data.frame(region = fit$HT$region,
                              time = fit$HT$time,
                              mean = fit$HT$HT.est.original,
                              type = "Direct"))
nmr <- rbind(nmr, data.frame(region = fit$smooth$region,
                              time = fit$smooth$time,
                              mean = fit$smooth$mean.original,
                              type = "Smoothed"))
nmr$time <- levels(data$time)[nmr$time]
nmr$time <- factor(nmr$time, levels = levels(data$time))
```

We can see the effect of temporal smoothing from the line plot below. Notice however, that, many direct estimates are \(0\) in this small simulated dataset, effectively not providing any information to the smoothing as \(\hat{V_i}\) is undefined in such cases. The corresponding smoothed estimates will then be estimated based entirely on the information from nearby regions and time periods.

This is a problem for smoothed direct estimates, and it is more severe as we try to smooth direct estimates at finer spatial and/or temporal resolution. In the next session, we will use a different type model-based estimates to overcome this issue. Nevertheless, this example serves its purpose as an illustration of space-time smoothing.

```
ggplot(nmr, aes(x = time, y = mean, color = type, group = type)) + geom_point() +
    geom_line() + facet_wrap(~region, ncol = 4) + theme(legend.position = "bottom")
```



Space-time smoothing: U5MR

As a final example, we look at all five surveys in the simulated dataset, `DemoData`.

Horvitz-Thompson estimates

First, we obtain the Horvitz-Thompson estimators using `getDirectList` on the list of surveys. Similar as before, we need to specify the survey design. In this case, strata are specified in the `strata` column, and clusters are specified by both the cluster ID (`clusterid`) and household ID (`id`). We extract the year labels as well for easier future use.

```
years <- levels(DemoData[[1]]$time)
data_multi <- getDirectList(births = DemoData, years = years, regionVar = "region",
    timeVar = "time", clusterVar = "~clustid+id", ageVar = "age", weightsVar = "weights",
    geo.recode = NULL)
```

After obtaining the direct estimates, we first aggregate theminto a single set of estimates using the inverse design-based variances as the weights.

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

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

After combining the direct estimates from multiple surveys, we will discuss the smoothing model for the national and subnational estimates respectively in the next two sections. In each case, we also discuss and compare smoothing the temporal trends on the yearly and period scales.

National estimates

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.

```
years.all <- c(years, "15-19")
fit1 <- fitINLA(data = data, geo = NULL, Amat = NULL, year_label = years.all,
    rw = 2, is.yearly = FALSE)
```

We can also estimate the Random Walk 2 random effects on the yearly scale, with direct estimates calculated in 5-year intervals.

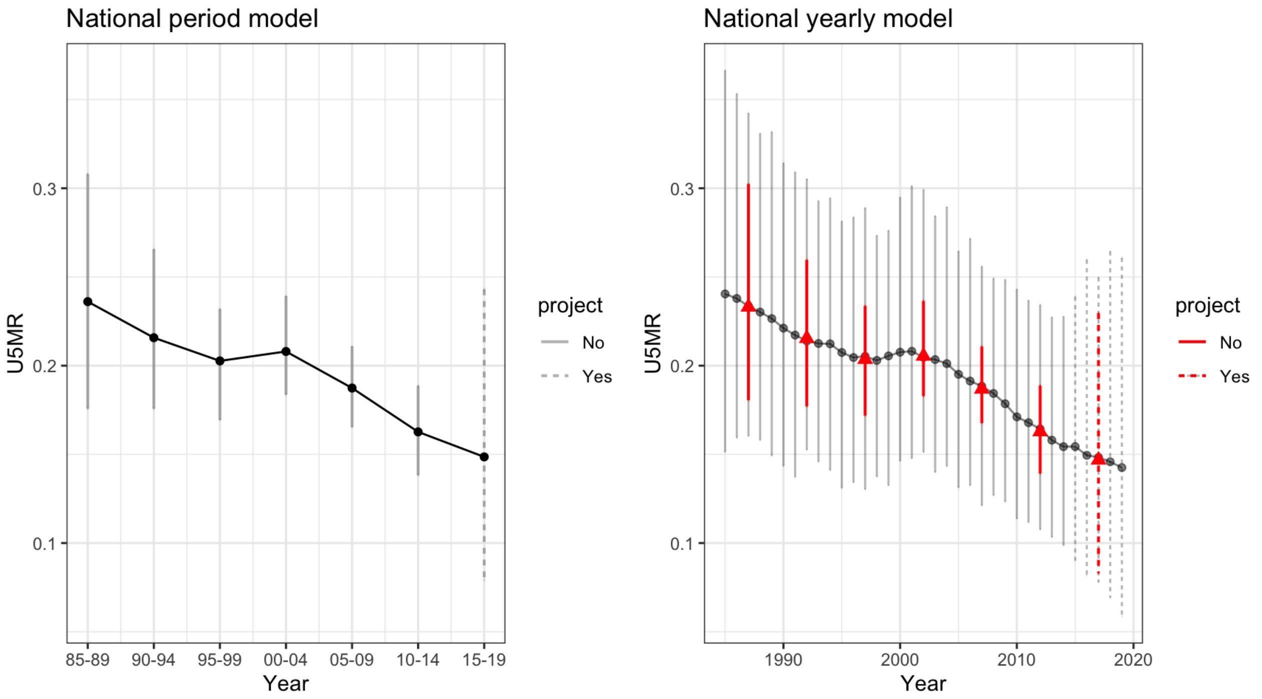
```
fit2 <- fitINLA(data = data, geo = NULL, Amat = NULL, year_label = years.all,
    year_range = c(1985, 2019), rw = 2, is.yearly = TRUE, m = 5)
```

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

```
out1 <- getSmoothed(fit1)
out2 <- getSmoothed(fit2)
```

We can compare the results visually using the function below.

```
g <- NULL
ylim <- range(c(out2$lower, out2$upper))
g[[1]] <- plot(out1, is.subnational=FALSE) + ggtitle("National period model") + ylim(ylim)
g[[2]] <- plot(out2, is.subnational=FALSE) + ggtitle("National yearly model") + ylim(ylim)
grid.arrange(grobs=g, ncol = 2)
```



Subnational estimates

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.

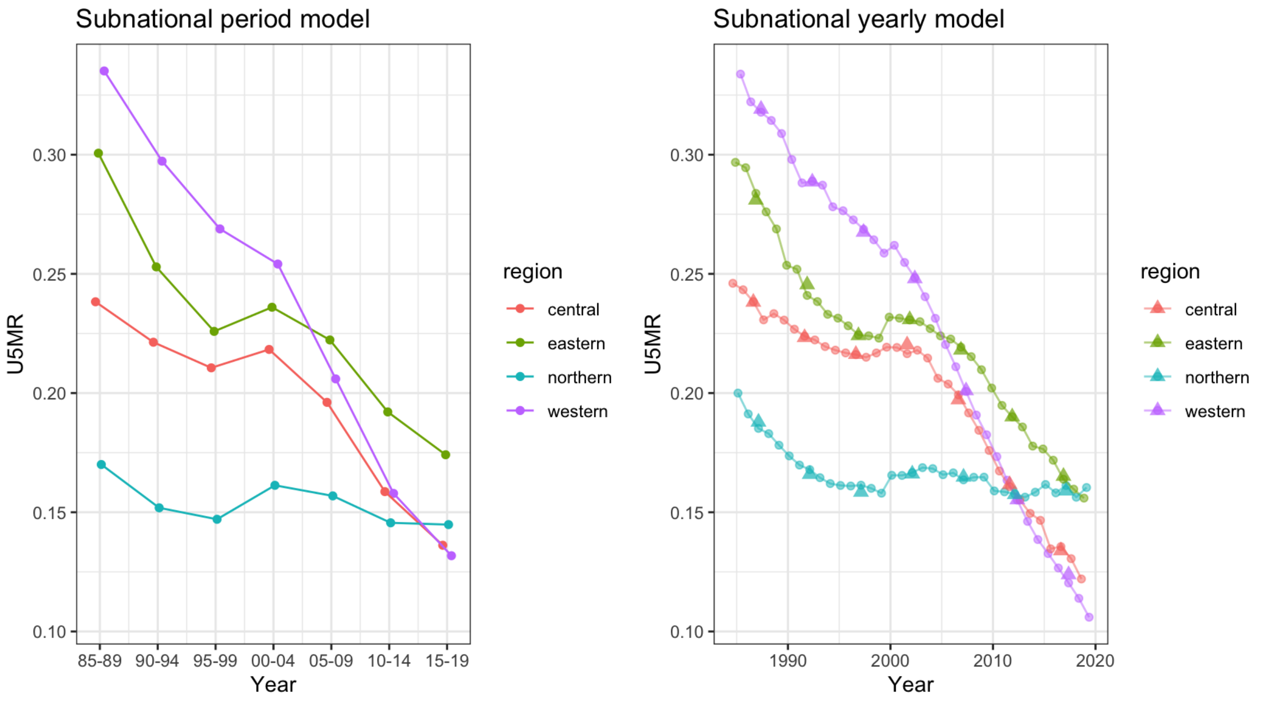
```
fit3 <- fitINLA(data = data, geo = DemoMap$geo, Amat = DemoMap$Amat, year_label = years.all,
    rw = 2, is.yearly = FALSE, type.st = 4)
out3 <- getSmoothed(fit3, Amat = DemoMap$Amat)
```

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

```
fit4 <- fitINLA(data = data, geo = DemoMap$geo, Amat = DemoMap$Amat, year_label = years.all,
    year_range = c(1985, 2019), rw = 2, is.yearly = TRUE, m = 5, type.st = 4)
out4 <- getSmoothed(fit4, Amat = DemoMap$Amat)
```

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

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

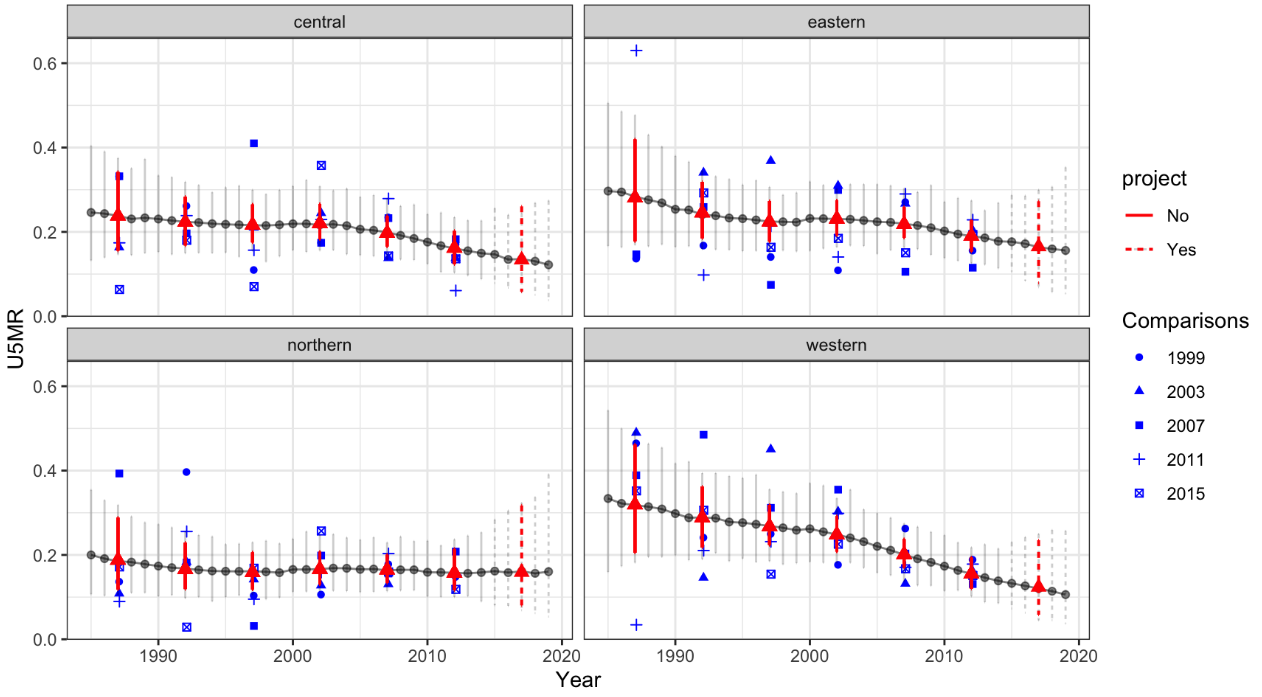


Subnational period model

Subnational yearly model

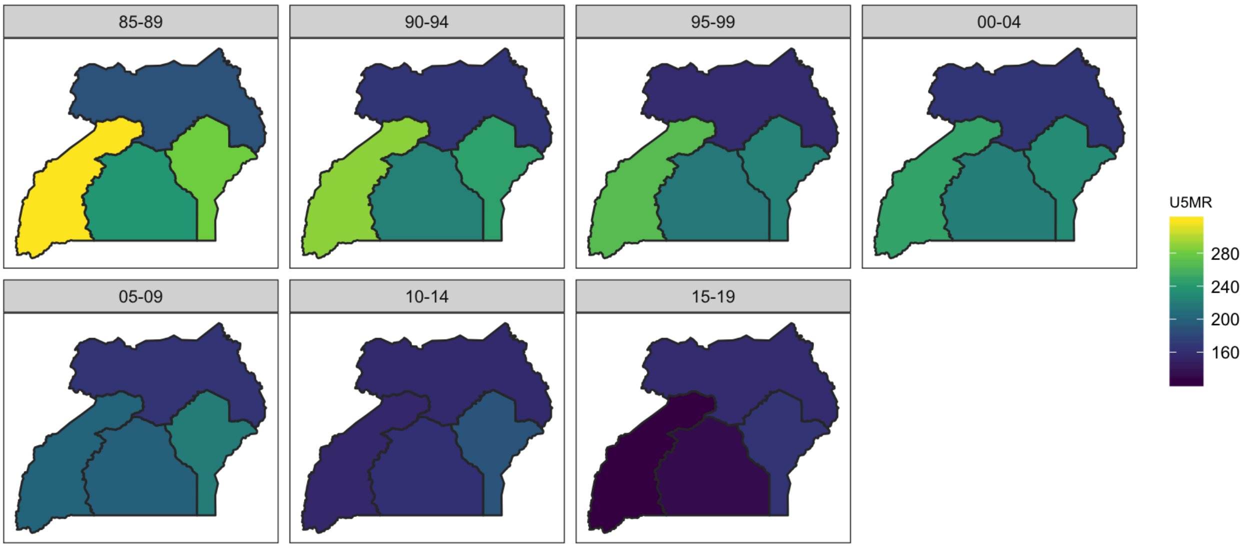
We can also add back the direct estimates for comparison when plotting the smoothed estimates.

```
plot(out4, is.yearly = TRUE, is.subnational = FALSE, data.add = data_multi,
    option.add = list(point = "mean", by = "surveyYears"), color.add = "blue",
    plot.CI = TRUE, alpha.CI = 0.2) + 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("median"), by.data = "region", by.geo = "NAME_final", is.long = TRUE,
    ncol = 4, per1000 = TRUE, legend.label = "U5MR")
```


85-89
90-94
95-99
00-04
05-09
10-14
15-19
U5MR
280
240
200
160