

Appendix for ‘Space-Time Smoothing of Survey Outcomes using the R Package SUMMER’

Zehang Richard Li Bryan D Martin Tracy Qi Dong Geir-Arne Fuglstad
 Jessica Godwin John Paige Andrea Riebler Samuel J Clark
 Jon Wakefield

Contents

A Prior specification	1
B Additional analysis of the simulated dataset	2
B.1 Unstratified cluster-level model	2
B.2 Stratified cluster-level model	3
C Obtaining and pre-processing Malawi DHS data	7
C.1 Loading DHS data using the <code>rdhs</code> package	8
C.2 Cleaning the DHS data	9
D Additional analysis of the 2010 and 2015 – 2016 Malawi DHS	11
D.1 Direct estimates	11
D.2 Space-time Fay-Harriot estimates	12
D.3 Cluster-level model estimation	13

A Prior specification

In all the model implementations, we apply penalised complexity (PC) priors to model the random effects (Simpson et al., 2017). These priors are proper and parameterization invariant. The basis of PC priors is to regard each model component as a flexible extension of a so-called base model. Considering an unstructured iid model component, the base model would be to remove this component from the linear predictor by letting its variance parameter go to zero. This is also the base model for any simple Gaussian model component with mean zero. The main idea is to follow Occam’s razor and favor less complex, or more intuitive, models unless the data suggest otherwise. Of note, state-of-the-art priors, such as the inverse gamma prior for a variance parameter, put zero density mass at the base model and as such do not allow the recovery of this model. The PC prior is specified by following a number of desirable principles and is derived based on the Kullback-Leibler distance of the flexible model from the base model. For details we refer to Simpson et al. (2017). Here, we will shortly comment on the PC priors relevant for the parameters used in our models and their default hyperparameters. All the PC priors can be specified by the user in the function calls using arguments such as `pc.u` and `pc.alpha`.

Considering a simple Gaussian model component with standard deviation parameter σ , the PC prior results in an exponential distribution for σ . The rate parameter λ_σ can be informed using a probability contrast of the form $\text{Prob}(\sigma > U_\sigma) = \alpha_\sigma$, which leads to $\lambda_\sigma = -\log(\alpha_\sigma)/U_\sigma$ (Simpson et al., 2017). The **SUMMER** package uses as default $U_\sigma = 1$ and $\alpha_\sigma = 0.01$, which means that the 99th percentile of the prior is at 1.

For the structured spatial random effects, we use the BYM2 model (Riebler et al., 2016; Simpson et al., 2017). It has a structured and unstructured term, and uses a single variance, σ^2 , that represents the marginal spatial variance and a mixing parameter $\phi \in [0, 1]$ specifying the proportion of spatial variation. To interpret σ as a marginal standard deviation, the spatial component needs to be scaled, so that $\text{Var}(e_i) \approx \text{Var}(S_i) \approx 1$. This leads to:

$$\mathbf{e} + \mathbf{S} = \sigma(\sqrt{(1 - \phi)}\mathbf{e}^* + \sqrt{\phi}\mathbf{S}^*)$$

where \mathbf{e}^* is iid normally distributed with fixed variance equal to 1 and \mathbf{S}^* is the scaled ICAR model. We follow Riebler et al. (2016) and scale the ICAR component so that the geometric mean of the marginal variances of S_i is equal to 1. Note that we also apply this scaling procedure to all intrinsic model components, such as random walk of order 1 or 2 components (Sørbye and Rue, 2014), to ensure interpretability of the prior distributions assigned to their flexibility parameters. The BYM2 model has a two-stage base model, with the first implying the absence of any spatial effect by setting σ equal to zero, and the second by assuming $\phi = 0$ and therefore only unstructured spatial variation. For σ we use an exponential prior as outlined before. The prior for ϕ depends on the study-specific neighborhood graph and is not available in closed form, see Riebler et al. (2016) for details. Its hyperparameter λ_ϕ can be derived from $\text{Prob}(\phi < U_\phi) = \alpha_\phi$. The **SUMMER** package uses as default $U_\phi = 0.5$ and $\alpha_\phi = 2/3$, which means that the 66.6th percentile of the prior is at 0.5, so that values of ϕ less than 0.5 are preferred a little more, a priori.

For the autoregressive model for time effects, we again use an exponential prior for the marginal standard deviation. For the autocorrelation correlation coefficient ω , we assume as base model $\omega = 1$. This represents a limiting random walk which assumes that the process does not change in time. The prior for ω is again not available in closed form, see Sørbye and Rue (2017) for details. Its hyperparameter λ_ω can be found from $\text{Prob}(\omega > U_\omega) = \alpha_\omega$. The **SUMMER** package uses as default $U_\omega = 0.7$ and $\alpha_\omega = 0.9$, which means that the 10th percentile of the prior is at 0.7, and therefore preferring values of ϕ that are close to 1.

The space-time interaction terms, δ_{it} , are modeled with the Type I, II, III, IV models of Knorr-Held (2000). The Type I model assumes iid interaction terms, the Type II model that the interactions are temporally structured but independent in space and the Type III model that the interactions are iid in time but spatially structured via an ICAR model. For the default Type IV interaction, we assume the specified temporal model and spatial (ICAR) structured effects interact. When the temporal component in the space-time interaction terms are modeled with a random walk of order 1 or an autoregressive model of order 1, we may also allow area-specific deviations from the main temporal trends by letting $\delta_{it} = b_i t + \delta_{it}^*$, where δ_{it}^* follows the specified interaction model and b_i are random slopes. This allows us to capture more flexible temporal dynamics, and may aid in area-specific predictions. The random slopes are modeled with a Gaussian prior. To facilitate interpretation, we scale the time index to be from -0.5 to 0.5 , so that the random slope can be interpreted as the total deviation from the main time trend from the first and last years to be projected, on the logit scale. Users can specify priors for the random slopes with the PC prior so that $\text{Prob}(|b| < U_b) = \alpha_b$ using arguments `pc.st.slope.u` and `pc.st.slope.alpha`.

B Additional analysis of the simulated dataset

B.1 Unstratified cluster-level model

In this section, we present additional analysis for the simulated dataset in Example 2 of the main paper. We first load the package, data, and compute the direct estimates of U5MR following the example in the main paper.

```
library(SUMMER)
library(stringr)
library(dplyr)
library(ggplot2)
library(patchwork)
data(DemoData)

periods <- c("85-89", "90-94", "95-99", "00-04", "05-09", "10-14")
directU5 <- getDirectList(births = DemoData, years = periods,
  regionVar = "region", timeVar = "time",
  clusterVar = "~clustid + id", ageVar = "age",
  weightsVar = "weights")
```

We now describe the fitting of the cluster level model. For simplicity, we first assume the survey was designed so that each of the four regions was a strata (thus no additional stratification within regions). The stratified analysis is left to the next section.

First, we calculate the number of person-months and number of deaths for each cluster, time period, and

age group. Notice that we do not need to impute all the 0's for combinations that do not exist in the data. We first create the data frame using the `getCounts()` function. The `getCounts()` function prepares the aggregated count dataset without modifying the original column names. For the model fitting functions to correctly identify the data columns, we rename the cluster ID and time period columns to be 'cluster' and 'years'. The response variable is 'Y' and the binomial total is 'total'.

```
counts.all <- NULL
for(i in 1:length(DemoData)){
  vars <- c("clustid", "region", "time", "age")
  counts <- getCounts(DemoData[[i]][, c(vars, "died")], variables = 'died',
                      by = vars, drop=TRUE)
  counts <- counts %>% mutate(cluster = clustid, years = time, Y=died)
  counts$survey <- names(DemoData)[i]
  counts.all <- rbind(counts.all, counts)
}
head(counts.all)

##   clustid region time age died total cluster years Y survey
## 1      36 central 85-89  0    0     1      36 85-89 0  1999
## 2      38 central 85-89  0    0     1      38 85-89 0  1999
## 3      91 central 85-89  0    0     1      91 85-89 0  1999
## 4     101 central 85-89  0    0     1     101 85-89 0  1999
## 5     128 central 85-89  0    0     1     128 85-89 0  1999
## 6     129 central 85-89  0    0     1     129 85-89 0  1999
```

With the created data frame, we fit the cluster-level model using the `smoothCluster()` function and obtain the estimates with the `getSmoothed()` function. Notice that here we need to specify the age groups (`age.groups`), the length of each age group (`age.n`) in months, and how the age groups are mapped to the temporal random effects (`age.rw.group`). In the default case, `age.rw.group = c(1, 2, 3, 3, 3, 3)` means the first two age groups each has its own temporal trend, the the following four age groups share the same temporal trend. We start with the default temporal model of random walk or order 2 on the 5-year periods in this dataset (with real data, we can use a finer temporal resolution). We add survey iid effects to the model as well using `survey.effect = TRUE` argument.

```
fit.bb.sim <- smoothCluster(data = counts.all, Amat = DemoMap$Amat,
                           family = "betabinomial",
                           year.label = c(periods, "15-19"),
                           age.group = c("0", "1-11", "12-23", "24-35", "36-47", "48-59"),
                           age.n = c(1, 11, 12, 12, 12, 12),
                           age.time.group = c(1, 2, 3, 3, 3, 3),
                           time.model = "rw2",
                           st.time.model = "ar1",
                           pc.st.slope.u = 2, pc.st.slope.alpha = 0.1,
                           survey.effect = TRUE)
est.bb.sim <- getSmoothed(fit.bb.sim, nsim = 1000)
```

We visualize the results from the cluster level model in Figure 1. We also overlay the survey-specific direct estimates using the `data.add` argument.

```
plot(est.bb.sim$overall, plot.CI=TRUE, data.add = directU5,
     option.add = list(point = "mean", by = "surveyYears"),
     color.add="steelblue") + facet_wrap(~region, ncol = 4)
```

B.2 Stratified cluster-level model

We now describe the fitting of the cluster-level model for U5MR taking into account the urban/rural stratification. For this simulated dataset, the strata variable is coded as region crossed by urban/rural status. For our analysis with urban/rural stratified model, we first construct a new strata variable that contains only the urban/rural status, i.e., the additional stratification within each region and computes the counts of person-months and death similar to the previous case.

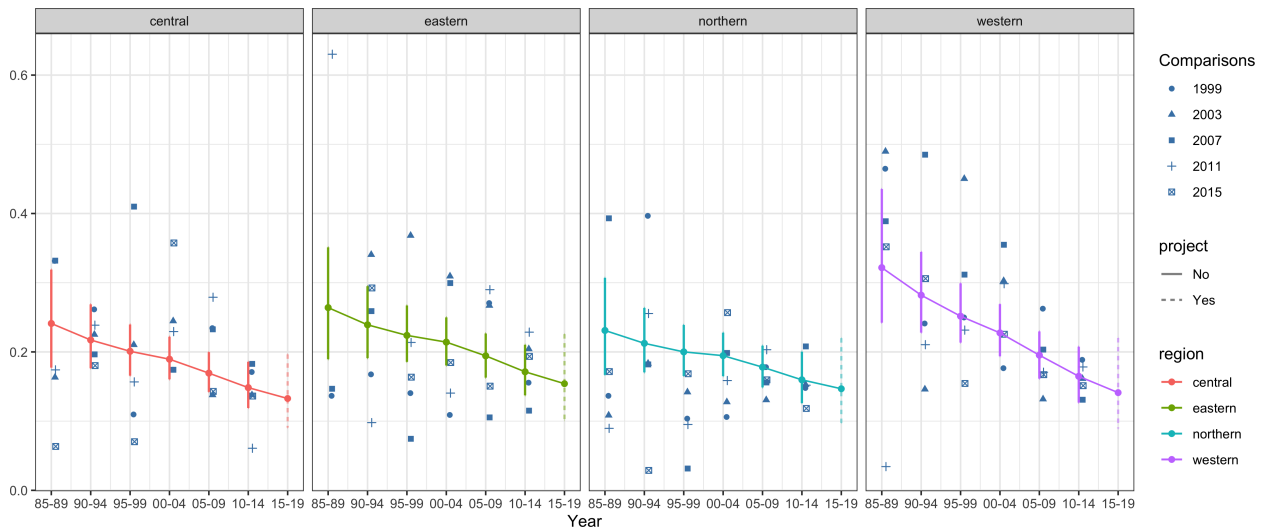


Figure 1: Cluster-level models estimates of subnational U5MR using multiple simulated surveys. The blue dots are direct estimates from each of the surveys.

```
for(i in 1:length(DemoData)){
  strata <- DemoData[[i]]$strata
  DemoData[[i]]$strata[grepl("urban", strata)] <- "urban"
  DemoData[[i]]$strata[grepl("rural", strata)] <- "rural"
}
counts.all <- NULL
for(i in 1:length(DemoData)){
  vars <- c("clustid", "region", "strata", "time", "age")
  counts <- getCounts(DemoData[[i]][, c(vars, "died")], variables = 'died',
    by = vars, drop=TRUE)
  counts <- counts %>% mutate(cluster = clustid, years = time, Y=died)
  counts$survey <- names(DemoData)[i]
  counts.all <- rbind(counts.all, counts)
}
```

With the created data frame, we fit the cluster-level model using the `smoothCluster` function. The temporal main effects are defined for each stratum separately (specified by `strata.time.effect = TRUE`, so in total six random walks are used to model the main temporal effect.

```
fit.bb <- smoothCluster(data = counts.all, Amat = DemoMap$Amat,
  family = "betabinomial",
  year.label = c( periods, "15-19"),
  age.group = c("0", "1-11", "12-23", "24-35", "36-47", "48-59"),
  age.n = c(1, 11, 12, 12, 12, 12),
  age.time.group = c(1, 2, 3, 3, 3, 3),
  time.model = "rw2",
  st.time.model = "ar1",
  pc.st.slope.u = 2, pc.st.slope.alpha = 0.1,
  survey.effect = TRUE,
  strata.time.effect = TRUE)
```

```
## -----
## Cluster-level model
## Main temporal model:      rw2
## Number of time periods:   7
## Spatial effect:          bym2
## Number of regions:       4
## Interaction temporal model: ar1
## Interaction type:         4
```

```

## Interaction random slopes: yes
## Number of age groups: 6
## Stratification: yes
## Number of age-specific fixed effect intercept per stratum: 6
## Number of age-specific trends per stratum: 3
## Strata-specific temporal trends: yes
## Survey effect: yes
## -----

est.bb <- getSmoothed(fit.bb, nsim = 1000, CI = 0.95, save.draws=TRUE)

## Starting posterior sampling...
## Cleaning up results...
## No strata weights has been supplied. Overall estimates are not calculated.

summary(fit.bb)

## -----
## Cluster-level model
## Main temporal model:          rw2
## Number of time periods:      7
## Spatial effect:              bym2
## Number of regions:           4
## Interaction temporal model:  ar1
## Interaction type:             4
## Interaction random slopes:   yes
## Number of age groups:        6
## Stratification:              yes
## Number of age group fixed effect intercept per stratum: 6
## Number of age-specific trends per stratum: 3
## Strata-specific temporal trends: yes
## Survey effect:               yes
## -----
## Fixed Effects
##               mean    sd 0.025quant 0.5quant 0.97quant mode kld
## age.intercept0:rural   -3.0 0.14      -3.3    -3.0     -2.7 -3.0  0
## age.intercept1-11:rural -4.7 0.11      -4.9    -4.7     -4.5 -4.7  0
## age.intercept12-23:rural -5.8 0.14      -6.1    -5.8     -5.5 -5.8  0
## age.intercept24-35:rural -6.6 0.20      -6.9    -6.6     -6.2 -6.6  0
## age.intercept36-47:rural -6.9 0.23      -7.3    -6.9     -6.4 -6.9  0
## age.intercept48-59:rural -7.3 0.29      -7.9    -7.3     -6.8 -7.3  0
## age.intercept0:urban    -2.7 0.13      -3.0    -2.7     -2.5 -2.7  0
## age.intercept1-11:urban  -5.0 0.13      -5.2    -5.0     -4.7 -5.0  0
## age.intercept12-23:urban -5.7 0.17      -6.1    -5.7     -5.4 -5.7  0
## age.intercept24-35:urban -7.1 0.29      -7.6    -7.1     -6.5 -7.1  0
## age.intercept36-47:urban -7.6 0.37      -8.3    -7.6     -6.9 -7.6  0
## age.intercept48-59:urban -8.0 0.46      -8.9    -8.0     -7.1 -8.0  0
##
## Slope fixed effect index:
## time.slope.group1: 0:rural
## time.slope.group2: 1-11:rural
## time.slope.group3: 12-23:rural, 24-35:rural, 36-47:rural, 48-59:rural
## time.slope.group4: 0:urban
## time.slope.group5: 1-11:urban
## time.slope.group6: 12-23:urban, 24-35:urban, 36-47:urban, 48-59:urban
## -----
## Random Effects
##           Name           Model
## 1  time.struct      RW2 model
## 2 time.unstruct      IID model

```

```
## 3 region.struct      BYM2 model
## 4   region.int Besags ICAR model
## 5   st.slope.id      IID model
## 6   survey.id       IID model
## -----
## Model hyperparameters
##
##                                mean      sd 0.025quant 0.5quant
## overdispersion for the betabinomial observations  0.002    0.001    0.001    0.002
## Precision for time.struct                        91.454  111.521    11.674    58.458
## Precision for time.unstruct                      831.467 3076.794    15.769   225.650
## Precision for region.struct                     252.997  671.939     5.264    88.691
## Phi for region.struct                          0.346    0.240     0.027    0.297
## Precision for region.int                       771.901 3485.913     7.931   168.610
## Group PACF1 for region.int                      0.893    0.203     0.253    0.970
## Precision for st.slope.id                       24.590   63.442     0.801    9.168
##
##                                0.97quant    mode
## overdispersion for the betabinomial observations  0.005  0.002
## Precision for time.struct                        349.819 27.909
## Precision for time.unstruct                     4750.233 35.610
## Precision for region.struct                     1367.096 11.338
## Phi for region.struct                          0.849  0.077
## Precision for region.int                       4531.658 15.399
## Group PACF1 for region.int                      0.999  1.000
## Precision for st.slope.id                       130.531  1.920
##
##                                [,1]
## log marginal-likelihood (integration) -3455
## log marginal-likelihood (Gaussian)    -3449
```

The `est.bb` object above computes the U5MR estimates by urban/rural strata. In order to obtain the overall region-specific U5MR, we need additional information on the population fractions of each stratum within regions. For illustration purpose, here we simulate the population totals over the years and use these population totals to compute the population fractions later.

```
pop.base <- expand.grid(region = c("central", "eastern", "northern", "western"),
                        strata = c("urban", "rural"))
pop.base$population <- round(runif(dim(pop.base)[1], 1000, 20000))
periods.all <- c(periods, "15-19")
pop <- NULL
for(i in 1:length(periods.all)){
  tmp <- pop.base
  tmp$population <- pop.base$population + round(rnorm(dim(pop.base)[1], mean = 0, sd = 200))
  tmp$years <- periods.all[i]
  pop <- rbind(pop, tmp)
}
head(pop)

##      region strata population years
## 1 central  urban    16529 85-89
## 2 eastern  urban    10630 85-89
## 3 northern urban    14664 85-89
## 4 western  urban    11470 85-89
## 5 central  rural     5836 85-89
## 6 eastern  rural     5479 85-89
```

In order to compute the aggregated estimates, we need the proportion of urban/rural populations within each region in each time period, as computed below in the `weight.strata` object.

```
weight.strata <- expand.grid(region = c("central", "eastern", "northern", "western"),
                             years = periods.all)
weight.strata$urban <- weight.strata$rural <- NA
for(i in 1:dim(weight.strata)[1]){
```

```

which.u <- which(pop$region == weight.strata$region[i] &
  pop$years == weight.strata$years[i] &
  pop$strata == "urban")
which.r <- which(pop$region == weight.strata$region[i] &
  pop$years == weight.strata$years[i] &
  pop$strata == "rural")
weight.strata[i, "urban"] <- pop$population[which.u] /
  (pop$population[which.u] + pop$population[which.r])
weight.strata[i, "rural"] <- 1 - weight.strata[i, "urban"]
}
head(weight.strata)

##      region years rural urban
## 1 central 85-89  0.26  0.74
## 2 eastern 85-89  0.34  0.66
## 3 northern 85-89  0.53  0.47
## 4 western 85-89  0.34  0.66
## 5 central 90-94  0.26  0.74
## 6 eastern 90-94  0.36  0.64

```

Now we can recompute the smoothed estimates with the population fractions.

```

est.bb <- getSmoothed(fit.bb, nsim = 1000, CI = 0.95, save.draws = TRUE,
  weight.strata = weight.strata)
head(est.bb$overall)

##      region years time area variance median mean upper lower rural urban is.yearly
## 5 central 85-89     1     1  0.00149   0.22 0.22  0.30  0.16  0.26  0.74    FALSE
## 6 central 90-94     2     1  0.00060   0.20 0.20  0.25  0.15  0.26  0.74    FALSE
## 7 central 95-99     3     1  0.00033   0.18 0.18  0.22  0.15  0.25  0.75    FALSE
## 1 central 00-04     4     1  0.00026   0.17 0.17  0.21  0.14  0.25  0.75    FALSE
## 2 central 05-09     5     1  0.00027   0.16 0.16  0.20  0.13  0.26  0.74    FALSE
## 3 central 10-14     6     1  0.00043   0.15 0.15  0.19  0.11  0.26  0.74    FALSE
##      years.num
## 5           NA
## 6           NA
## 7           NA
## 1           NA
## 2           NA
## 3           NA

```

We can compare the stratum-specific and aggregated U5MR estimates now.

```

g1 <- plot(est.bb$stratified, plot.CI = TRUE) + facet_wrap(~strata) + ylim(0, 0.5)
g2 <- plot(est.bb$overall, plot.CI = TRUE) + ylim(0, 0.5) + ggtitle("Aggregated estimates")
g1 + g2

```

C Obtaining and pre-processing Malawi DHS data

In this section, we present the full workflow of downloading and pre-processing the two most recent Malawi DHS datasets, the 2010 and 2015 – 2016 DHS. We also describe how the count data used in the main paper is produced. First, we load and process the spatial polygon data.

```

data(MalawiMap)
MalawiGraph <- getAmat(MalawiMap, names=MalawiMap$ADM2_EN)
mapPlot(geo=MalawiMap, by.geo = "ADM2_EN")

```

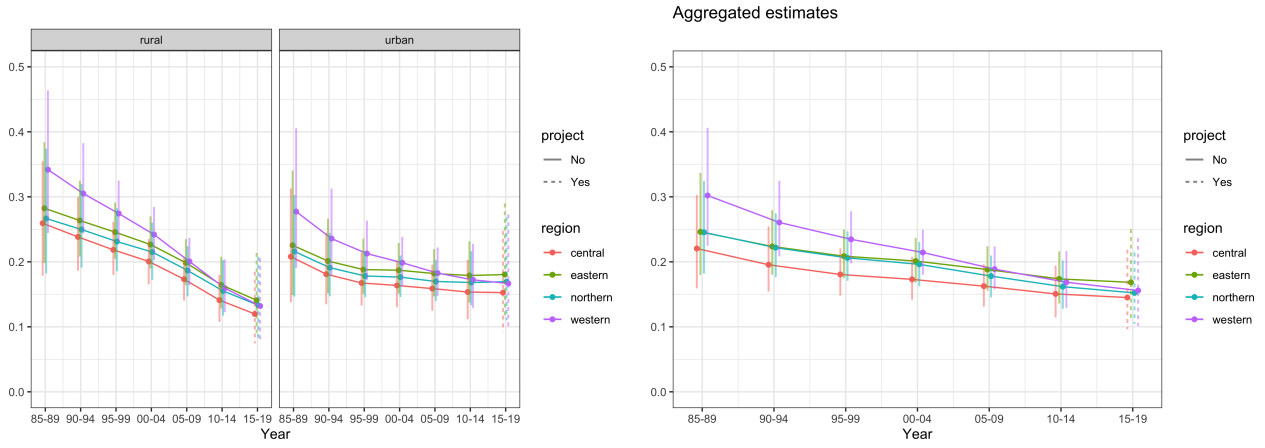


Figure 2: Comparing stratum-specific estimates of U5MR and the aggregated U5MR using simulated population weights.



Figure 3: Admin-2 regions in Malawi.

C.1 Loading DHS data using the `rdhs` package

In this example, we use the 2010 and 2015–2016 Malawi DHS surveys. The DHS website (<https://dhsprogram.com/data/available-datasets.cfm?ctryid=24>) provides links to download all the surveys with registration. Once access is approved, we can use the `rdhs` package to load data directly from the DHS API (Watson and Eaton, 2019). We first download both the birth records (BR) and GPS data (GE) for these two surveys. Notice the following codes can only be run after registration with DHS and set up the credentials using the `rdhs` package. Additional information on this step can be found in the documentation of the `rdhs` package.


```

library(rdhs)
sv <- dhs_surveys(countryIds = "MW", surveyType = "DHS",
  surveyYearStart = 2010)
BR <- dhs_datasets(surveyIds = sv$SurveyId, fileFormat = "Flat",
  fileType = "BR")
BRfiles <- get_datasets(BR$FileName, reformat=TRUE)
GPS <- dhs_datasets(surveyIds = sv$SurveyId, fileFormat = "Flat",
  fileType = "GE")
GPSfiles <- get_datasets(GPS$FileName, reformat=TRUE)

The downloaded data can then be loaded by the returned file paths.

Surv2010 <- readRDS(BRfiles[[1]])
Surv2015 <- readRDS(BRfiles[[2]])
DHS2010.geo <- readRDS(GPSfiles[[1]])
DHS2015.geo <- readRDS(GPSfiles[[2]])

load("data/downloaded.rda")

```

C.2 Cleaning the DHS data

We then use the `getBirths()` function to process the raw birth history into person-month format. We label the person-month records with 6 age groups specified by `month.cut`. In this example, instead of using 5 year periods, we work directly with yearly estimates specified by `year.cut`.

```

DHS2010 <- getBirths(data = Surv2010,
  month.cut = c(1, 12, 24, 36, 48, 60),
  year.cut = seq(2000, 2020, by = 1), strata = "v022")
DHS2015 <- getBirths(data = Surv2015,
  month.cut = c(1, 12, 24, 36, 48, 60),
  year.cut = seq(2000, 2020, by = 1), strata = "v022")

```

We perform similar processing steps for the 2015–2016 DHS survey birth records. Since only a small fraction of observations are available in 2016, we remove the partial year observations in this survey.

```
DHS2015 <- subset(DHS2015, time != 2016)
```

The DHS dataset does not usually have sufficient spatial resolution information in the birth records file, so we need to use the GPS datasets of cluster locations to assign records to the admin-2 areas. This process can be highly survey-specific and require more extensive data manipulation. In the 2010 DHS dataset, the GPS file contains the cluster ID (DHSCLUST), urbanicity indicator (URBAN_RURA), and DHSCLUST variable in the format of “admin-2__urbanrural”, with some spelling and capitalization differences as in the polygon file. From the GPS dataset, we processed a list of clusters and their corresponding admin-2 areas.

```

cluster.list <- data.frame(DHS2010.geo) %>%
  distinct(DHSCLUST, DHSREGNA, URBAN_RURA) %>%
  mutate(admin2 = gsub(" - rural", "", DHSREGNA)) %>%
  mutate(admin2 = gsub(" - urban", "", admin2)) %>%
  mutate(admin2 = recode(admin2, "nkhatabay" = "nkhata bay")) %>%
  mutate(admin2 = recode(admin2, "nkhota kota" = "nkhatakota")) %>%
  mutate(admin2 = str_to_title(admin2)) %>%
  mutate(urban = ifelse(URBAN_RURA=="U", 1, 0)) %>%
  select(v001 = DHSCLUST, admin2, urban)

head(cluster.list, n=3)

##   v001 admin2 urban
## 1    1  Dedza    0
## 2    2 Balaka    0
## 3    3 Mchinji    0

```

We then merge this list to the main person-month file, which adds the `admin2` and `urban` columns to the data frame.

```
DHS2010 <- DHS2010 %>% left_join(cluster.list)
head(DHS2010)
```

```
##      dob survey_year died id.new      caseid v001 v002 v004      v005 v021 v022      v023
## 1 1316          NA    0      1      1 1 2      1      1      1 1902867      1      12 central
## 2 1316          NA    0      1      1 1 2      1      1      1 1902867      1      12 central
## 3 1316          NA    0      1      1 1 2      1      1      1 1902867      1      12 central
## 4 1316          NA    0      1      1 1 2      1      1      1 1902867      1      12 central
## 5 1316          NA    0      1      1 1 2      1      1      1 1902867      1      12 central
## 6 1316          NA    0      1      1 1 2      1      1      1 1902867      1      12 central
##      v024 v025      v139 bidx agemonth obsStart obsStop obsmonth year  age time strata
## 1 central rural central      1      0      1316      1316      1316 109      0 2009      12
## 2 central rural central      1      1      1317      1317      1317 109 1-11 2009      12
## 3 central rural central      1      2      1318      1318      1318 109 1-11 2009      12
## 4 central rural central      1      3      1319      1319      1319 109 1-11 2009      12
## 5 central rural central      1      4      1320      1320      1320 109 1-11 2009      12
## 6 central rural central      1      5      1321      1321      1321 110 1-11 2010      12
##      admin2 urban
## 1 Dedza      0
## 2 Dedza      0
## 3 Dedza      0
## 4 Dedza      0
## 5 Dedza      0
## 6 Dedza      0
```

We perform similar processing steps for the 2015–2016 DHS GPS data. To illustrate the idiosyncratic data processing steps required for each survey, the DHSClust variable in this dataset is in the form of “admin-1_urbanrural” and does not allow us to parse the admin-2 area names. The strata variable v022 in the birth records, however, is in the “admin-2_urbanrural” format. Therefore, we first merge the v022 variables to the GPS data and proceed in a similar fashion to the previous example.

```
cluster.list <- data.frame(DHS2015.geo) %>%
  mutate(v001 = DHSClust) %>%
  left_join(DHS2015, by = "v001") %>%
  distinct(v001, v022, URBAN_RURA) %>%
  mutate(admin2 = gsub(" - rural", "", v022)) %>%
  mutate(admin2 = gsub(" - urban", "", admin2)) %>%
  mutate(admin2 = recode(admin2, "nkhatabay" = "nkhata bay")) %>%
  mutate(admin2 = recode(admin2, "nkhota kota" = "nkhatakota")) %>%
  mutate(admin2 = str_to_title(admin2)) %>%
  mutate(urban = ifelse(URBAN_RURA=="U", 1, 0)) %>%
  select(v001, admin2, urban)
DHS2015 <- DHS2015 %>% left_join(cluster.list)
head(DHS2015)
```

```
##      dob survey_year died id.new      caseid v001 v002 v004      v005 v021      v022
## 1 1334          NA    0      1      1 12 2      1      1      1 118748      1 ntchisi - urban
## 2 1334          NA    0      1      1 12 2      1      1      1 118748      1 ntchisi - urban
## 3 1334          NA    0      1      1 12 2      1      1      1 118748      1 ntchisi - urban
## 4 1334          NA    0      1      1 12 2      1      1      1 118748      1 ntchisi - urban
## 5 1334          NA    0      1      1 12 2      1      1      1 118748      1 ntchisi - urban
## 6 1334          NA    0      1      1 12 2      1      1      1 118748      1 ntchisi - urban
##      v023      v024 v025      v139 bidx agemonth obsStart obsStop
## 1 ntchisi - urban central region urban central region      1      0      1334      1334
## 2 ntchisi - urban central region urban central region      1      1      1335      1335
## 3 ntchisi - urban central region urban central region      1      2      1336      1336
## 4 ntchisi - urban central region urban central region      1      3      1337      1337
## 5 ntchisi - urban central region urban central region      1      4      1338      1338
## 6 ntchisi - urban central region urban central region      1      5      1339      1339
##      obsmonth year  age time strata admin2 urban
```

```
## 1      1334  111    0 2011 ntchisi - urban Ntchisi      1
## 2      1335  111 1-11 2011 ntchisi - urban Ntchisi      1
## 3      1336  111 1-11 2011 ntchisi - urban Ntchisi      1
## 4      1337  111 1-11 2011 ntchisi - urban Ntchisi      1
## 5      1338  111 1-11 2011 ntchisi - urban Ntchisi      1
## 6      1339  111 1-11 2011 ntchisi - urban Ntchisi      1
```

Then we use the `getCounts()` function to obtain the count data format input and stack the two DHS survey data into a single data frame.

```
vars <- c("v001", "v025", "admin2", "time", "age", "v005")
dat1 <- getCounts(DHS2010[, c(vars, "died")], variables = 'died',
                  by = vars, drop=TRUE)
dat1$survey = "DHS2010"
dat2 <- getCounts(DHS2015[, c(vars, "died")], variables = 'died',
                  by = vars, drop=TRUE)
dat2$survey = "DHS2015"
DHS.counts <- rbind(dat1, dat2) %>%
  mutate(cluster = v001, strata = v025, region = admin2,
         years = time, Y = died)
```

The analysis in the main paper is carried out only on the 2015–2016 dataset, which is obtained using the same procedure.

D Additional analysis of the 2010 and 2015 – 2016 Malawi DHS

In this section, we provide additional workflow of estimating U5MR from 2000 to 2019 using the two DHS surveys. The focus of this section is on the models not discussed in details in Example 3 of the main paper.

D.1 Direct estimates

We use `getDirectList` to obtain direct estimates from both surveys and combine into the ‘meta-analysis’ estimator using `aggregateSurvey`. Notice that the `survey` variable in the returned data frame contains a numeric index of each survey in the list, and the `surveyYears` contains the survey names we assigned in the input list.

```
direct <- getDirectList(births = list(DHS2010=DHS2010, DHS2015=DHS2015),
                      years = 2000:2019, regionVar = "admin2", timeVar = "time",
                      clusterVar = "~v001 + v002", ageVar = "age", weightsVar = "v005")
direct.comb <- aggregateSurvey(direct)
```

When additional information is available to adjust the direct estimates from the surveys, we use the methods described in Li et al. (2019). We can perform the ratio adjustment to the direct estimates using the `getAdjusted()` function. For the two surveys in Malawi, the calculated HIV adjustment ratios as described in Walker et al. (2012) are stored in `MalawiData$HIV.yearly`. In order to get the correct uncertainty bounds, we also need to specify the columns corresponding to the unadjusted uncertainty bounds, the `lower` and `upper` columns, which are on the probability scale in this case.

```
data(MalawiData)
direct.2010 <- subset(direct, survey == 1)
direct.2010.hiv <- getAdjusted(data = direct.2010,
                              ratio = subset(MalawiData$HIV.yearly, survey == "DHS2010"),
                              logit.lower = NA, logit.upper = NA,
                              prob.lower = "lower", prob.upper = "upper")
direct.2015 <- subset(direct, survey == 2)
direct.2015.hiv <- getAdjusted(data = direct.2015,
                              ratio = subset(MalawiData$HIV.yearly, survey == "DHS2015"),
                              logit.lower = NA, logit.upper = NA,
                              prob.lower = "lower", prob.upper = "upper")
```

Finally, we combine the direct estimates into `direct.comb.hiv`.

```
direct.hiv <- rbind(direct.2010.hiv, direct.2015.hiv)
direct.comb.hiv <- aggregateSurvey(direct.hiv)
```

D.2 Space-time Fay-Harriot estimates

We now fit a national Fay-Harriot model with the calculated direct estimates using `smoothDirect()` and `getSmoothed()` functions.

```
fit.national.unadj <- smoothDirect(data = direct.comb, Amat = NULL,
                                   year.label = 2000:2019, year.range = c(2000, 2019),
                                   time.model = "rw2", m = 1)
est.unadj <- getSmoothed(fit.national.unadj)
```

For comparison, we smooth both the unadjusted direct estimates and the direct estimates with HIV adjustments.

```
fit.national.hiv <- smoothDirect(data = direct.comb.hiv, Amat = NULL,
                                  year.label = 2000:2019, year.range = c(2000, 2019),
                                  time.model = "rw2", m = 1)
est.hiv <- getSmoothed(fit.national.hiv)
```

In addition, we also demonstrate the benchmarking procedure described in Li et al. (2019), where we first fit a smoothing model and then benchmark the results with the UN IGME estimates – the latter are based on more extensive data (Alkema and New, 2014). We first calculate the adjustment ratio compared to the 2019 UN IGME estimates.

```
UN <- MalawiData$IGME2019
UN.est <- UN$mean[match(2000:2019, UN$years)]
Smooth.est <- est.hiv$median[match(2000:2019, est.hiv$years)]
UN.adj <- data.frame(years = 2000:2019, ratio = Smooth.est / UN.est)
head(UN.adj, n = 3)

##   years ratio
## 1  2000   1.1
## 2  2001   1.2
## 3  2002   1.1
```

We then fit the smoothing model on the benchmarked direct estimates.

```
direct.comb.benchmark <- getAdjusted(data = direct.comb.hiv, ratio = UN.adj,
                                     logit.lower = NA, logit.upper = NA,
                                     prob.lower = "lower", prob.upper = "upper")
fit.benchmark <- smoothDirect(data = direct.comb.benchmark, Amat = NULL,
                              year.label = 2000:2019, year.range = c(2000, 2019),
                              time.model = "rw2", m = 1)
est.benchmark <- getSmoothed(fit.benchmark)
```

We compare the different Fay-Harriot estimates in Figure 4. Compared to the raw estimates, HIV adjustments lead to higher estimates in the earlier years. The benchmarking step produces a national trend that follows the same trajectory as the UN IGME estimates.

```
g1 <- plot(est.unadj, is.subnational=FALSE, proj.year = 2016) +
  ggtitle("Unadjusted") + ylim(c(0, 0.22))
g2 <- plot(est.hiv, is.subnational=FALSE, proj.year = 2016) +
  ggtitle("With HIV adjustment") + ylim(c(0, 0.22))
g3 <- plot(est.benchmark, is.subnational=FALSE, proj.year = 2016, data.add = UN,
          option.add = list(point = "mean"), label.add = "UN", color.add = "red") +
  ggtitle("Benchmarked to UN IGME") + ylim(c(0, 0.22))
g1 + g2 + g3
```

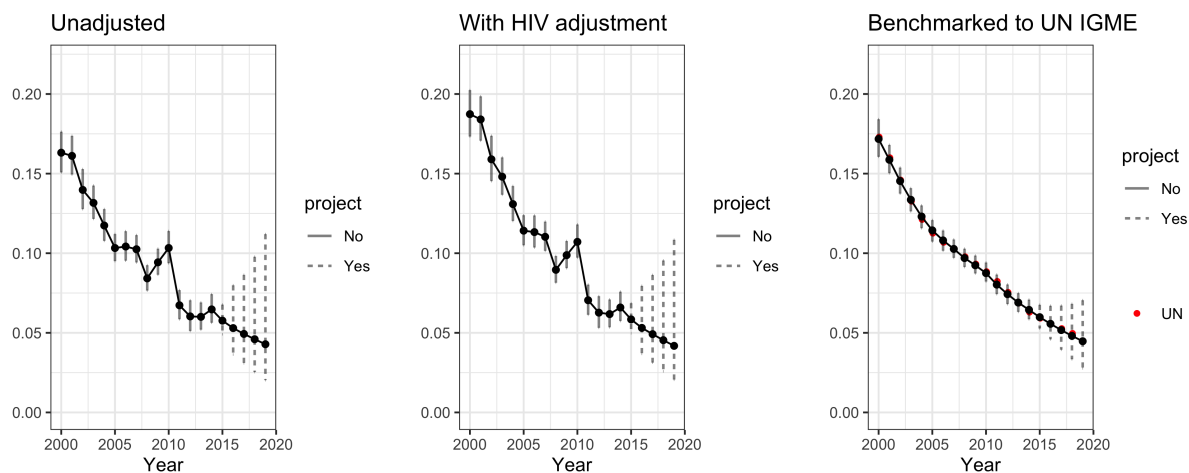


Figure 4: Comparison of the different Fay-Harriot estimates. The UN IGME estimates are plotted as red dots on the benchmarked plot.

Finally, the subnational models can be fit in the same manner. Since the direct estimates are already on the yearly scale, we do not need to perform the transformation from periods to years. So we set `is.yearly` to `FALSE` to fit the period model directly.

```
fit.smooth.direct <- smoothDirect(data = direct.comb.benchmark,
  Amat = MalawiGraph,
  year.label = 2000:2019, year.range = c(2000, 2019),
  time.model = "rw2", m = 1,
  type.st = 4, pc.alpha = 0.05, pc.u = 1)
est.smooth.direct <- getSmoothed(fit.smooth.direct)
```

D.3 Cluster-level model estimation

We now describe the fitting of the cluster level model using the two DHS surveys. With the created data frame, we first fit the national model with survey-year-specific HIV adjustment factors specified using `bias.adj` and `bias.adj.by` arguments. The adjustments are performed as offsets in the likelihood as described before. We also add a sum-to-zero survey effect term.

```
fit.bb.nat <- smoothCluster(data = DHS.counts, Amat = NULL,
  family = "betabinomial", year.label = 2000:2019,
  time.model = "rw2",
  bias.adj = MalawiData$HIV.yearly,
  bias.adj.by = c("years", "survey"),
  survey.effect = TRUE)
```

The `getSmoothed` function then produces estimates from the fitted cluster-level model. Similar to before, we take `nsim` draws of the posterior distribution to calculate the summaries of the U5MR estimates.

```
est.bb.nat <- getSmoothed(fit.bb.nat, nsim = 1000, save.draws = TRUE)
```

In this example, no strata weights are provided and thus the overall estimates are empty. Given a data frame of strata proportions, we can rerun the `getSmoothed` function to re-aggregate the stratified estimates. The `save.draws` argument in the `getSmoothed` call allows the raw posterior draws to be returned as part of the output object. This can be helpful in such situations, as posterior draws already computed can be inserted into new `getSmoothed` calls using the `draws` argument to avoid resampling again.

Figure 5 shows the national estimates of U5MR in Malawi for urban and rural stratum respectively using the cluster-level model.

```
plot(est.bb.nat$stratified, is.subnational=FALSE, proj.year = 2016) + facet_wrap(~strata)
```

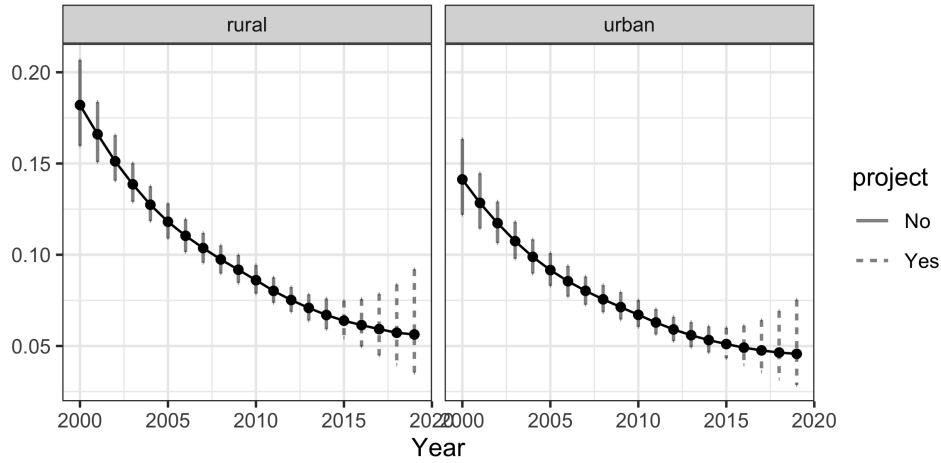


Figure 5: National estimated rural and urban U5MR in Malawi using DHS from 2010 and 2015–2016.

We can also fit the subnational model in the same fashion. Additional analysis can be similarly carried out as in the previous example with simulated data. For the cluster-level model, benchmarking to national estimates is also implemented in the package using the procedure described in [okonek2022computationally] with additional information on population fractions by region. We refer readers to the package vignette for more details.

```
fit.bb <- smoothCluster(data = DHS.counts, Amat = MalawiGraph,
  family = "betabinomial",
  year.label = 2000:2019,
  time.model = "rw2", st.time.model = "ar1",
  pc.st.slope.u = 2,
  pc.st.slope.alpha = 0.1,
  bias.adj = MalawiData$HIV.yearly,
  bias.adj.by = c("years", "survey"),
  survey.effect = TRUE,
  strata.time.effect = TRUE)
est.bb <- getSmoother(fit.bb, nsim = 1000, save.draws = TRUE)
```

The U5MR by strata can be visualized directly. Notice that in order to produce the overall estimates by region and time, additional information on population fractions in urban/rural is necessary, similar to the analysis conducted in the main paper using simulated data. For more details on obtaining such factions, we refer readers to [fuglstad2021two] and [wu2021spatial].

```
plot(est.bb$stratified) + facet_wrap(~strata)
```

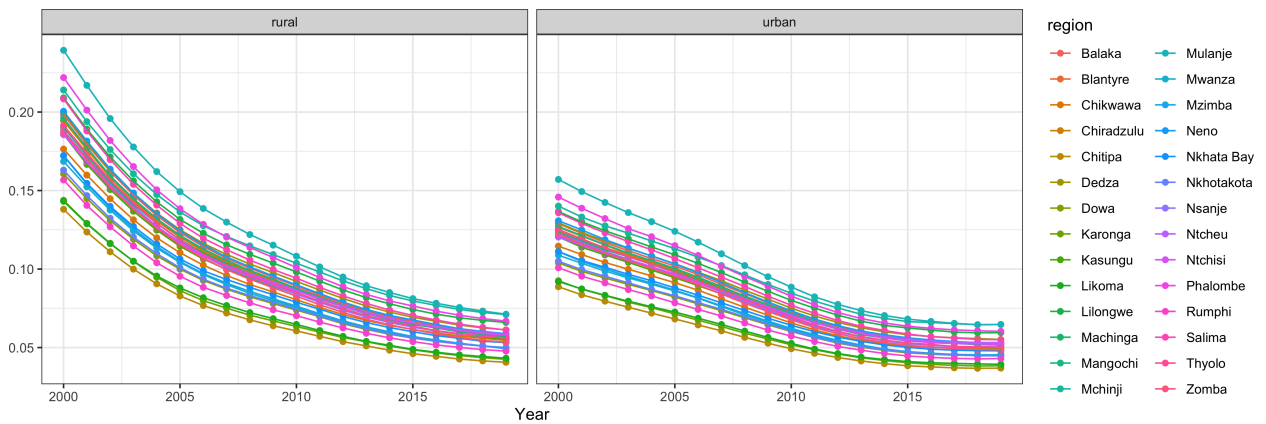


Figure 6: Subnational estimated rural and urban U5MR in Malawi using DHS from 2010 and 2015–2016.

References

- Alkema, L. and New, J. R. (2014). Global estimation of child mortality using a Bayesian B-spline bias-reduction model. *The Annals of Applied Statistics*, 8:2122–2149.
- Knorr-Held, L. (2000). Bayesian modelling of inseparable space-time variation in disease risk. *Statistics in Medicine*, 19:2555–2567.
- Li, Z. R., Hsiao, Y., Godwin, J., Martin, B. D., Wakefield, J., and Clark, S. J. (2019). Changes in the spatial distribution of the under five mortality rate: small-area analysis of 122 DHS surveys in 262 subregions of 35 countries in Africa. *PLoS One*. Published January 22, 2019.
- Riebler, A., Sørbye, S. H., Simpson, D., and Rue, H. (2016). An intuitive Bayesian spatial model for disease mapping that accounts for scaling. *Statistical Methods in Medical Research*, 25:1145–1165.
- Simpson, D., Rue, H., Riebler, A., Martins, T., and Sørbye, S. (2017). Penalising model component complexity: A principled, practical approach to constructing priors (with discussion). *Statistical Science*, 32:1–28.
- Sørbye, S. H. and Rue, H. (2014). Scaling intrinsic gaussian markov random field priors in spatial modelling. *Spatial Statistics*, 8:39–51.
- Sørbye, S. H. and Rue, H. (2017). Penalised complexity priors for stationary autoregressive processes. *Journal of Time Series Analysis*, 38(6):923–935.
- Walker, N., Hill, K., and Zhao, F. (2012). Child mortality estimation: methods used to adjust for bias due to AIDS in estimating trends in under-five mortality. *PLoS Med*, 9:e1001298.
- Watson, O. and Eaton, J. (2019). **rdhs**: API client and dataset management for the Demographic and Health Survey (DHS) data. R package version 0.6.3.