Spatio-temporal Under-five Mortality Methods for Estimation

Load Package and 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`.

First, we load the package and load the necessary data. INLA is not in a standard repository, so we check if it is available and install it if it is not.

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

data(DemoData)
data(DemoMap)
```

`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")
```

Make Country Summary

Next, we obtain Horvitz-Thompson estimators using `countrySummary_mult`.

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

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

Read Maps

In this step, we separate the output from `read_shape` to use as function arguments.

```
    geo <- DemoMap$geo
    mat <- DemoMap$Amat
```

Make Priors

Using our adjacency matrix, we 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.

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

Prepare data for meta analysis

Before fitting the model, we first aggregate estimators from different surveys.

```
dim(data)
```

```
## [1] 150  10
```

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

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

Fit INLA Model for national estimates

Now we are ready to fit the models. The codes to perform the new model fitting is attached at the end of this documentation.

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".

Period model

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_names = years.all,
                year_range = c(1985, 2019), priors = priors, rw = 2,
                is.yearly=FALSE, m = 5)
```

Yearly model

Similarly as before, we can estimate the Random Walk 2 random effects on the yearly scale.

```
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)
```

```
## Warning in inla.model.properties.generic(inla.trim.family(model), (mm[names(mm) == : Model 'rgeneric
##   Use this model with extra care!!! Further warnings are disabled.
```

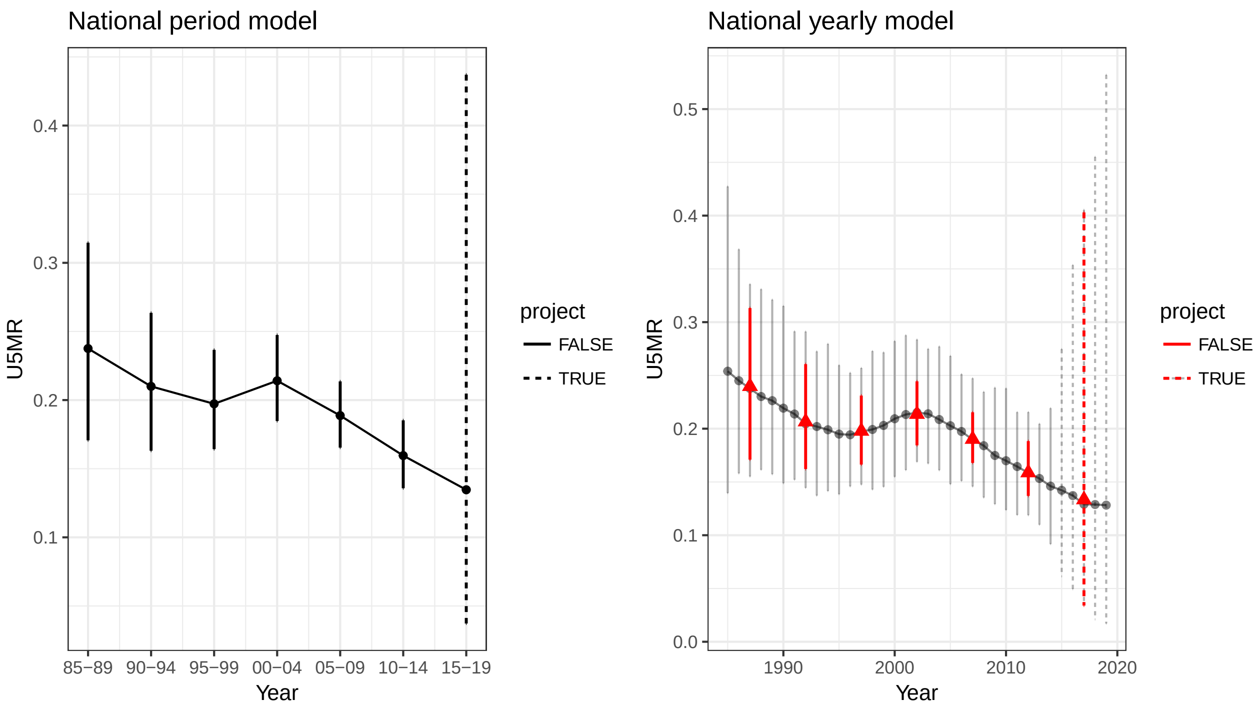
Obtain smoothed estimates

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

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

We can compare the results visually using the function below.

```
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)
```



Fit INLA model for subnational estimates

Similarly we can fit the full model on all subnational regions.

Period model

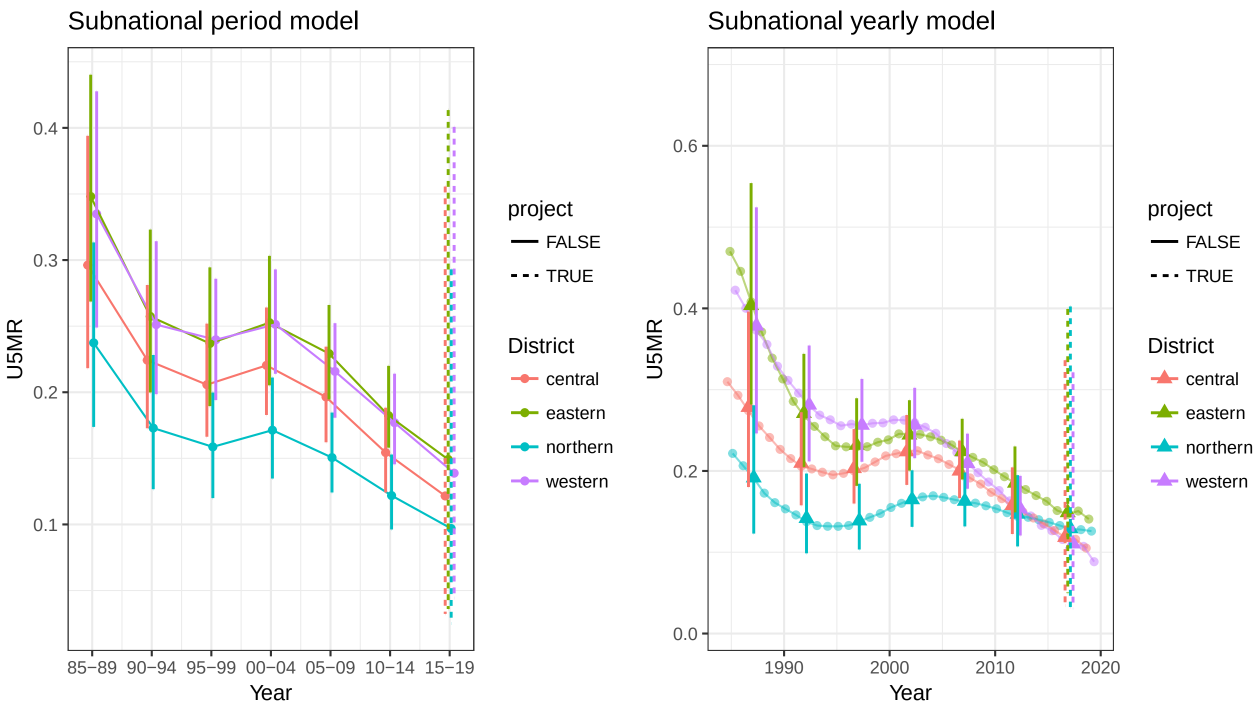
```
fit3 <- fitINLA(data = data, geo = geo, Amat = mat, year_names = years.all,
                year_range = c(1985, 2019), priors = priors, rw = 2,
                is.yearly=FALSE)
out3 <- projINLA(fit3, Amat = mat, is.yearly = FALSE)
```

Yearly model with type IV interaction

```
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)
```

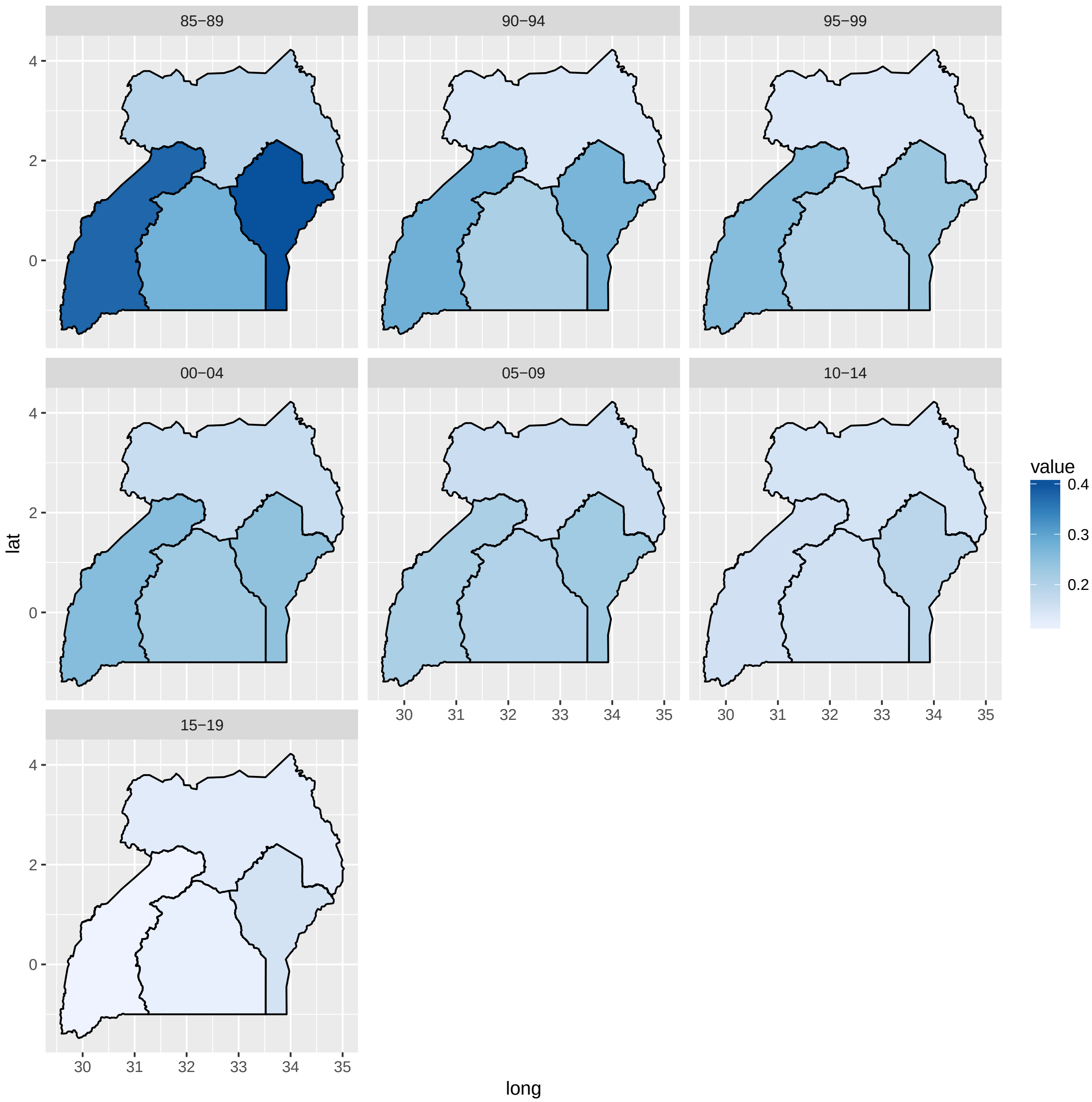
Compare plots

```
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)
```



Map visualization of changes across time

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



Simple spatial smoothing examples

In this section we show two simple spatial smoothing example using data created from the model survey data, and a Kenya Admin 1 map with 8 regions.

```
data(DemoData2)
data(DemoMap2)
```

The DemoData2 dataset contains the survey information and two response variables, age and tobacco usage.

```
head(DemoData2)
```

```
##   clustid id  region age  weights        strata tobacco.use
## 1       1  1 nairobi  30 1.057703 nairobi.urban           0
```

```
## 2        1   3 nairobi   22 1.057703 nairobi.urban           0
## 3        1   4 nairobi   42 1.057703 nairobi.urban           0
## 4        2   4  nyanza   25 1.057703  nyanza.urban           0
## 5        1   5 nairobi   25 1.057703 nairobi.urban           0
## 6        1   6 nairobi   37 1.057703 nairobi.urban           0
```

Normal model

We first generate some 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 from the ICAR random fields as follows

```r
set.seed(123)
sim.Q <- function(Q){
  eigenQ <- eigen(Q)
  rankQ <- qr(Q)$rank
  sim <- as.vector(eigenQ$vectors[,1:rankQ] %*%
            matrix(
              rnorm(rep(1, rankQ), rep(0, rankQ), 1/sqrt(eigenQ$values[1:rankQ])),
            ncol = 1))
  sim
}
Q <- DemoMap2$Amat * -1
diag(Q) <- 0
diag(Q) <- -1 * apply(Q, 2, sum)
struct.error <- sim.Q(Q)
```

We generate the mean height for each region by

```r
mu <- 70 + struct.error
regions <- colnames(DemoMap2$Amat)
```

We generate the data by

```r
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.

```r
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)
```

```
## FUN is not specified, default to be no transformation
```

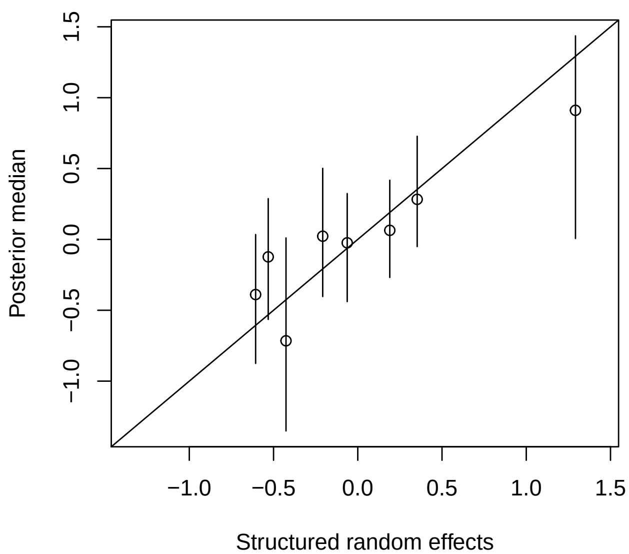
The posterior median of the structured random effects can be obtained from:

```
fit$fit$summary.random$reg.struct[, "0.5quant"]
```

```
## [1]  0.91062314  0.06395608  0.02256723  0.28264232 -0.71543457 -0.02419443
## [7] -0.12319878 -0.38904264
```

We can compare the posterior of the structured random effects with the truth values from the simulation.

```
lim <- range(c(struct.error, fit$fit$summary.random$reg.struct[, "0.025quant"],
               fit$fit$summary.random$reg.struct[, "0.975quant"]))
plot(struct.error, fit$fit$summary.random$reg.struct[, "0.5quant"],  xlab = "Structured random effects"
segments(x0 = struct.error, x1 = struct.error,
         y0 = fit$fit$summary.random$reg.struct[, "0.025quant"],
         y1 = fit$fit$summary.random$reg.struct[, "0.975quant"])
abline(c(0, 1))
```



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

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

```
##     HT.est     HT.sd       region
## 2 71.41449 0.2589442      nairobi
## 1 70.18577 0.1981691      central
## 4 70.26359 0.3135852        coast
## 3 70.61735 0.2736472      eastern
## 6 69.11042 0.3521545       nyanza
## 8 70.02198 0.2920432  rift valley
## 7 70.09129 0.2648998      western
## 5 69.59892 0.2658214 northeastern
```

The posterior summaries of $\mu_i|\hat{\mu}_i$ can be obtained by

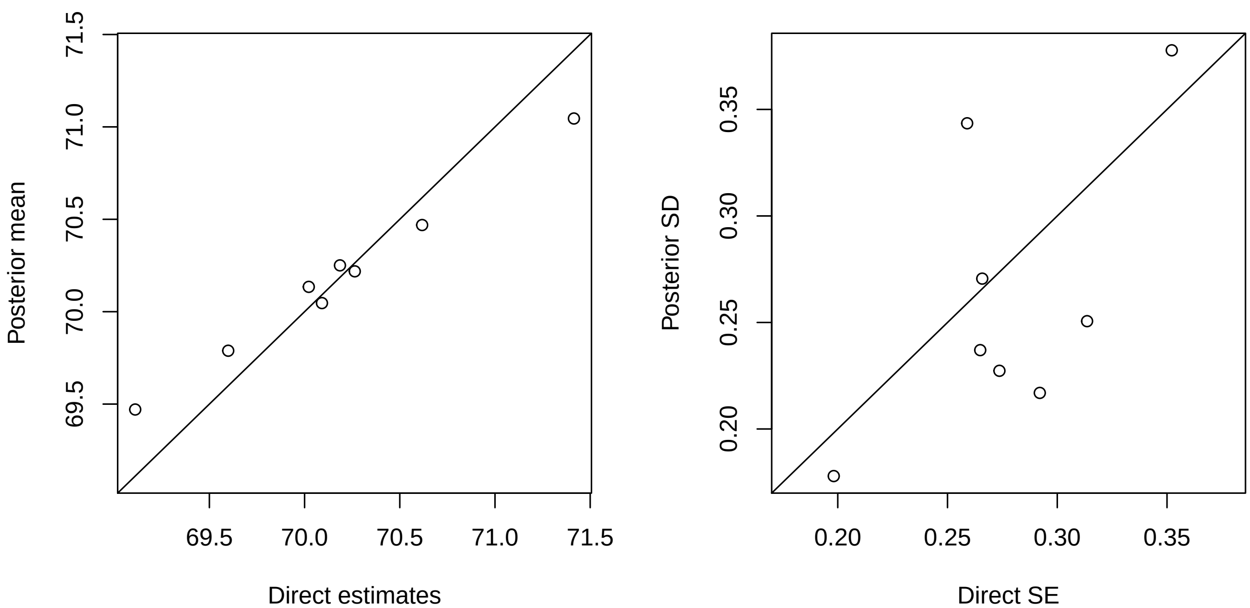
```
fit$smooth[, c("mean", "sd", "median", "lower", "upper", "region")]
```

```
##       mean         sd    median     lower     upper       region
```

```
## 1 71.04541 0.3434955 71.08062 70.20183 71.62917      nairobi
## 2 70.25065 0.1778905 70.25067 69.89622 70.60084      central
## 3 70.21840 0.2506011 70.21805 69.72027 70.72246        coast
## 4 70.46899 0.2273098 70.46048 70.05860 70.93503      eastern
## 5 69.47061 0.3777570 69.45338 68.76246 70.25802       nyanza
## 6 70.13441 0.2169330 70.14426 69.68556 70.54712  rift valley
## 7 70.04651 0.2370137 70.05317 69.57246 70.49749      western
## 8 69.78879 0.2705798 69.78373 69.26371 70.30345 northeastern
```

We can compare the direct and smoothed estimates and their standard errors.

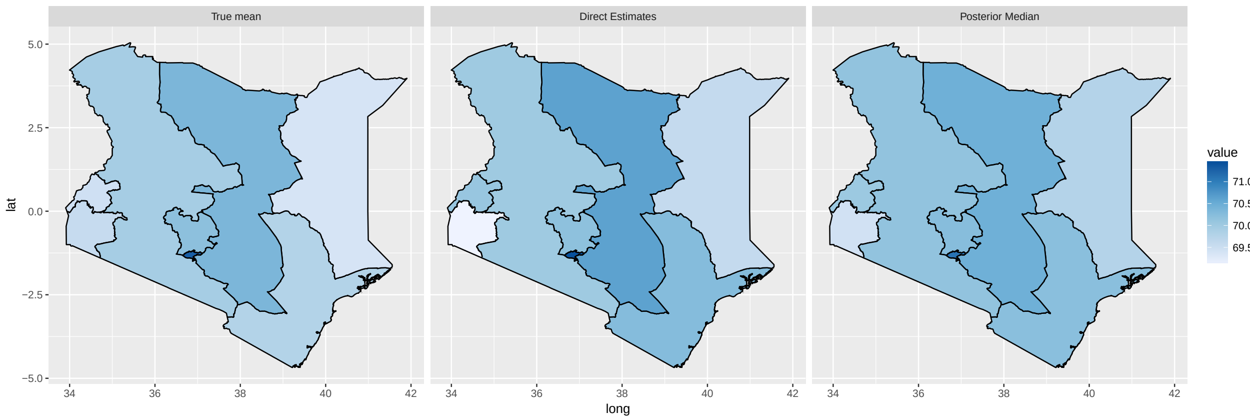
```
par(mfrow = c(1, 2))
lim <- range(c(fit$HT$HT.est, fit$smooth$mean))
plot(fit$HT$HT.est, fit$smooth$mean, xlim = lim, ylim = lim, xlab = "Direct estimates", ylab = "Posteri
abline(c(0, 1))
lim <- range(c(fit$HT$HT.sd, fit$smooth$sd))
plot(fit$HT$HT.sd, fit$smooth$sd, xlim = lim, ylim = lim, xlab = "Direct SE", ylab = "Posterior SD")
abline(c(0, 1))
```



We can also compare the direct and smoothed estimates on the map.

```
combined <- merge(fit$HT, fit$smooth, by = "region")
combined$truth <- mu[match(combined$region, regions)]
mapPlot(data = combined, geo = DemoMap2$geo,
        variables=c("truth", "HT.est", "median"),
        labels = c("True mean", "Direct Estimates", "Posterior Median"),
        by.data = "region", by.geo = "NAME_final", is.long=FALSE)
```

```
## Using region as id variables
```



Binary model with logistic link

Similar as in the previous subsection, suppose we denote the tobacco usage of observation k in area i to be y_{ik}, and the associated design weight to be w_{ik}. Under the design-based approach to inference, we can calculate the weighted estimator of prevalence to be

$$\hat{p}_i = \frac{\sum_k w_{ik} y_{ik}}{\sum_k w_{ik}}$$

and the associated variance $\widehat{var}(\hat{p}_i)$. By delta method, if we denote $\hat{\theta}_i = \log(\frac{\hat{p}_i}{1-\hat{p}_i})$, the asymptotic distribution of $\hat{\theta}_i$ is

$$\hat{\theta}_i \sim \text{Normal}(\log(\frac{p_i}{1-p_i}), \hat{V}_i), \quad \hat{V}_i = \frac{\widehat{var}(\hat{p}_i)}{(\hat{p}_i(1-\hat{p}_i))^2}$$

We then use INLA to fit the following Bayesian hierarchical model:

$$\begin{aligned}
\hat{\theta}_i &= \log\left(\frac{\hat{p}_i}{1-\hat{p}_i}\right) \sim \text{Normal}(\theta_i, \hat{V}_i) \\
\theta_i &= \beta + \epsilon_i + \delta_i, \\
\epsilon_i &\sim \text{Normal}(0, \sigma_\epsilon^2) \\
\delta_i &\sim \text{ICAR}(\sigma_\delta^2)
\end{aligned}$$

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="binomial", responseVar="tobacco.use",
                strataVar="strata", weightVar="weights", regionVar="region",
                clusterVar = "~clustid+id", hyper=NULL, CI = 0.95)
```

```
## FUN is not specified, default to be expit()
```

The direct estimates of the prevalence of tobacco usage, i.e., $\hat{p}_i$ accounting for survey design are

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

```
##   HT.est.original HT.variance.original       region
## 2      0.02748435         3.263723e-05      nairobi
## 1      0.04269545         6.222132e-05      central
## 4      0.07381327         7.796782e-05        coast
```

```
## 3      0.03453175         1.293472e-04      eastern
## 6      0.02809128         4.460260e-05       nyanza
## 8      0.07163454         2.165121e-04  rift valley
## 7      0.03462895         9.602971e-05      western
## 5      0.03735942         3.724319e-05 northeastern
```

The logit transformed direct estimates, $\hat{\theta}_i$ and the associated asymptotic standard deviations are

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

```
##      HT.est     HT.sd       region
## 2 -3.566270 0.2137345      nairobi
## 1 -3.110029 0.1929914      central
## 4 -2.529537 0.1291590        coast
## 3 -3.330734 0.3411316      eastern
## 6 -3.543803 0.2446150       nyanza
## 8 -2.561848 0.2212583  rift valley
## 7 -3.327823 0.2931361      western
## 5 -3.249095 0.1696911 northeastern
```

The posterior summaries of $\theta_i|\hat{\theta}_i$ can be obtained by

```
fit$smooth[, c("mean", "sd", "median", "lower", "upper", "region")]
```

```
##        mean        sd    median     lower     upper       region
## 1 -3.413011 0.1975402 -3.408581 -3.811719 -3.039909      nairobi
## 2 -3.111458 0.1683809 -3.110953 -3.443773 -2.781261      central
## 3 -2.615333 0.1321753 -2.615062 -2.875354 -2.356882        coast
## 4 -3.184440 0.2414236 -3.174090 -3.688738 -2.731562      eastern
## 5 -3.397220 0.2183763 -3.391888 -3.839304 -2.985165       nyanza
## 6 -2.783291 0.2004619 -2.792069 -3.150346 -2.369068  rift valley
## 7 -3.247065 0.2373924 -3.242015 -3.727277 -2.792064      western
## 8 -3.200087 0.1577831 -3.198592 -3.513472 -2.894907 northeastern
```

The posterior summaries of $p_i|\hat{\theta}_i$ can be obtained by

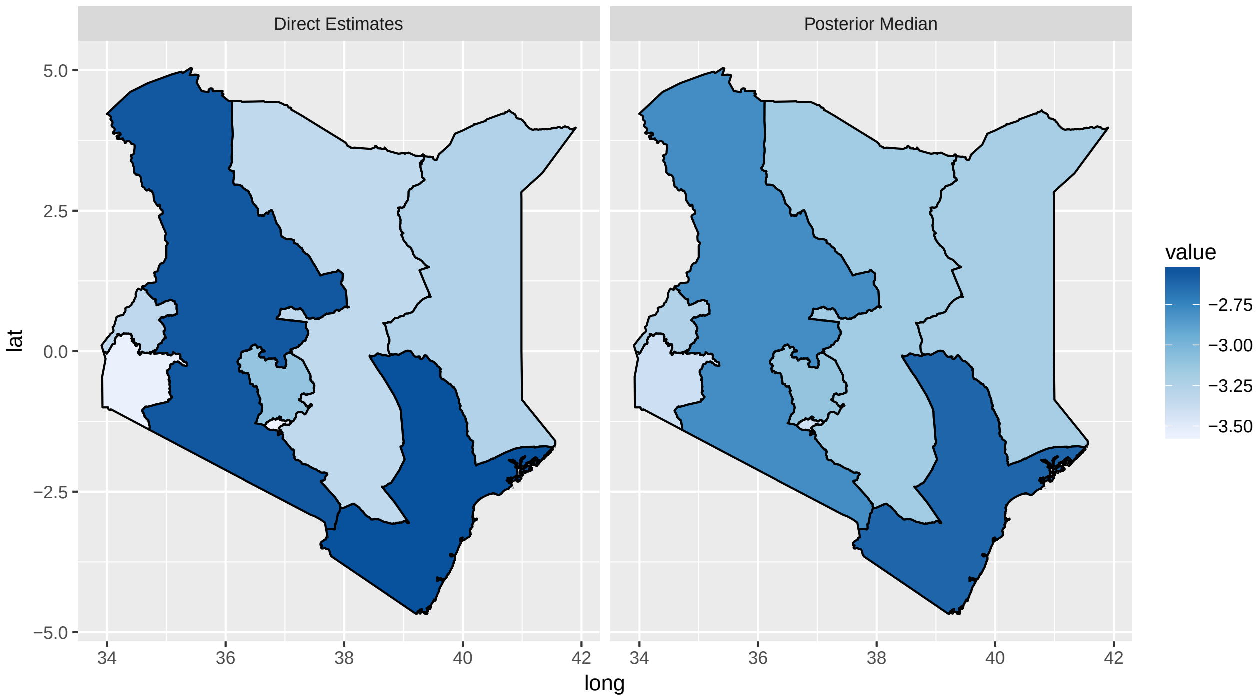
```
fit$smooth[, c("mean.trans", "sd.trans", "median.trans", "lower.trans", "upper.trans", "region")]
```

```
##   mean.trans    sd.trans median.trans lower.trans upper.trans       region
## 1 0.03239989 0.006131302   0.03198777  0.02158242  0.04557671      nairobi
## 2 0.04311575 0.006984147   0.04260846  0.03090433  0.05832735      central
## 3 0.06862602 0.008436211   0.06817033  0.05347034  0.08647247        coast
## 4 0.04075525 0.009336007   0.04012757  0.02436470  0.06115900      eastern
## 5 0.03305807 0.006944660   0.03254554  0.02099683  0.04804785       nyanza
## 6 0.05916138 0.011418730   0.05767767  0.04103424  0.08536118  rift valley
## 7 0.03829918 0.008702414   0.03756360  0.02339875  0.05751429      western
## 8 0.03959012 0.006018790   0.03920716  0.02893731  0.05244045 northeastern
```

We can now compare the direct and smoothed variables on the map.

```
combined <- merge(fit$HT, fit$smooth, by = "region")
mapPlot(data = combined, geo = DemoMap2$geo, variables=c("HT.est", "median"),
        labels = c("Direct Estimates", "Posterior Median"),
        by.data = "region", by.geo = "NAME_final", is.long=FALSE)
```

```
## Using region as id variables
```


Direct Estimates
Posterior Median
5.0
2.5
0.0
−2.5
−5.0
lat
34
36
38
40
42
long
value
−2.75
−3.00
−3.25
−3.50