2023 554 SUMMER Package R Notes

Jon Wakefield
Departments of Biostatistics and Statistics
University of Washington

2023-02-10

Small Area Estimation (SAE)

In these notes, SAE via the `SUMMER` package will be illustrated.

Details on `SUMMER`, including a vignette, can be found at https://cran.r-project.org/web/packages/SUMMER/index.html.

We illustrate with the Washington State BRFSS diabetes example and will obtain:

- Naive estimates
- Weighted estimates
- Estimates from a binomial BYM2 model
- Estimates from Fay-Herriot models

Load `SUMMER` package

We first load the `SUMMER` package.

This package also depends on `INLA`, so we need to make sure `INLA` is installed. Note that `INLA` is not on CRAN so it has a special installation process. Here, we check if `INLA` is available and install it if it is not.

```
library(SUMMER)
if (!isTRUE(requireNamespace("INLA", quietly = TRUE))) {
  install.packages("INLA",
                   repos = c(getOption("repos"),
                             INLA="https://inla.r-inla-download.org/R/stable"),
                   dep=TRUE)
}
```

Read in Data

`BRFSS` contains the full BRFSS dataset with 16,283 observations:

- `diab2` variable is the binary indicator of Type II diabetes
- `strata` is the strata indicator and
- `rwt_llcp` is the final weight.

For the purpose of this analysis, we first remove records with missing HRA code or diabetes status from this dataset.

```
data(BRFSS)
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.

```
library(sf)  # Load sf for spatial analysis
library(prioritizr)  # Allows us to create an adjacency matrix

data(KingCounty)
KingCounty <- st_as_sf(KingCounty)
mat <- adjacency_matrix(KingCounty)
colnames(mat) <- rownames(mat) <- KingCounty$HRA2010v2_
mat <- as.matrix(mat[1:dim(mat)[1], 1:dim(mat)[1]])
mat[1:2, 1:2]
##              Auburn-North Auburn-South
## Auburn-North            0            1
## Auburn-South            1            0
```

Create survey object

We load the survey package and then define the survey object for the BRFSS data. We have stratified, disproportionate sampling, so note the arguments:

- `weights`
- `strata`

We then calculate the direct (weighted) estimates.

```
library(survey)
design <- svydesign(ids = ~1, weights = ~rwt_llcp, strata = ~strata,
    data = BRFSS)
direct <- svyby(~diab2, ~hracode, design, svymean)
head(direct, n = 7)
##                                                  hracode      diab2         se
## Auburn-North                                Auburn-North 0.10403154 0.02147752
## Auburn-South                                Auburn-South 0.23293289 0.04897800
## Ballard                                          Ballard 0.07047572 0.02225241
## Beacon/Gtown/S.Park                  Beacon/Gtown/S.Park 0.08083033 0.02603522
## Bear Creek/Carnation/Duvall Bear Creek/Carnation/Duvall 0.05166773 0.01190146
## Bellevue-Central                        Bellevue-Central 0.05914082 0.01485885
## Bellevue-NE                                  Bellevue-NE 0.05772789 0.01509705
```

Binomial spatial smoothing model

We ignore the design and fit the model:

$$\begin{aligned} y_i | p_i &\sim \text{Binomial}(n_i, p_i) \\ \theta_i &= \log\left(\frac{p_i}{1-p_i}\right) = \alpha + b_i \end{aligned}$$

with b_i following a BYM2 model, i.e., an iid normal random effect and an intrinsic CAR (ICAR) random effect.

The binomial smoothing model is fitted by specifying NULL for the survey characteristics.

The `smoothSurvey` function

Note how the polygon information is input, and the neighbors in the `Amat` argument - this is required for the ICAR.

```r
smoothed <- smoothSurvey(data = BRFSS, geo = KingCounty, Amat = mat,
    responseType = "binary", responseVar = "diab2", strataVar = NULL,
    weightVar = NULL, regionVar = "hracode", clusterVar = NULL,
    CI = 0.95)
```

The usual INLA summaries can be found in `smoothed$fit`:

```r
smoothed$fit$summary.fixed
##                  mean         sd 0.025quant  0.5quant 0.975quant      mode
## (Intercept) -2.353603 0.03303594  -2.419006 -2.353481  -2.288934 -2.353278
##                      kld
## (Intercept) 1.431449e-09
smoothed$fit$summary.hyper
##                                 mean        sd 0.025quant   0.5quant
## Precision for region.struct 15.1709668 4.9499756  7.7539731 14.4056629
## Phi for region.struct        0.8558322 0.1378805  0.4888582  0.9009497
##                             0.975quant       mode
## Precision for region.struct 27.0305411 12.9899015
## Phi for region.struct        0.9955581  0.9923835
```

Now examine some of the other components:

```r
names(smoothed)
## [1] "HT"             "smooth"         "smooth.overall" "fit"
## [5] "CI"             "Amat"           "responseType"   "formula"
## [9] "msg"
names(smoothed$HT)
## [1] "region"        "HT.est"        "HT.var"        "HT.logit.est"
## [5] "HT.logit.var"  "HT.logit.prec" "n"             "y"
names(smoothed$smooth)
##  [1] "region"       "mean"         "var"          "median"       "lower"
##  [6] "upper"        "logit.mean"   "logit.var"    "logit.median" "logit.lower"
## [11] "logit.upper"
head(smoothed$HT, n = 4)
##               region     HT.est       HT.var HT.logit.est HT.logit.var
## 1        Auburn-North 0.14028777 0.0004338385    -1.812902   0.02982513
## 2        Auburn-South 0.23204420 0.0009845287    -1.196804   0.03100377
## 3             Ballard 0.06666667 0.0001121121    -2.639057   0.02895753
## 4 Beacon/Gtown/S.Park 0.08571429 0.0003731778    -2.367124   0.06076389
##   HT.logit.prec   n  y
## 1      33.52878 278 39
## 2      32.25414 181 42
## 3      34.53333 555 37
## 4      16.45714 210 18
```

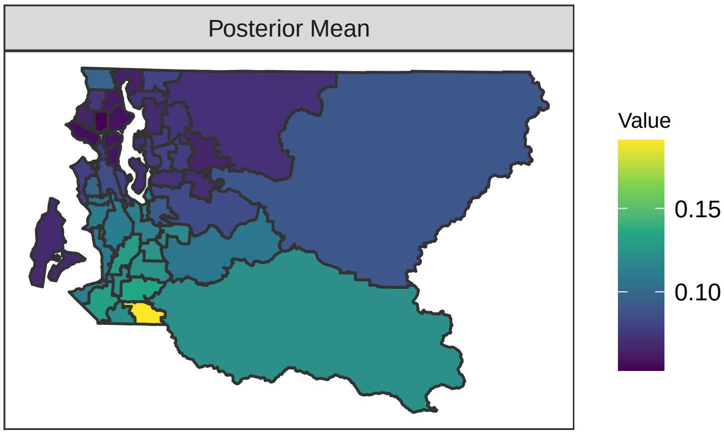
The smoothed estimates of p_i and θ_i can be found in the smooth object returned by the function, and the direct estimates are stored in the HT object (without specifying survey weights, these are the simple binomial probabilities).

```r
head(smoothed$smooth, n = 1)
##         region      mean          var    median     lower     upper logit.mean
## 1 Auburn-North 0.1352512 0.0002478173 0.1344345 0.1067145 0.1684375  -1.862901
##    logit.var logit.median logit.lower logit.upper
## 1 0.01800993    -1.863009   -2.124745   -1.599246
head(smoothed$HT, n = 1)
##         region    HT.est       HT.var HT.logit.est HT.logit.var HT.logit.prec
```

```
## 1 Auburn-North 0.1402878 0.0004338385    -1.812902    0.02982513       33.52878
##     n  y
## 1 278 39
```

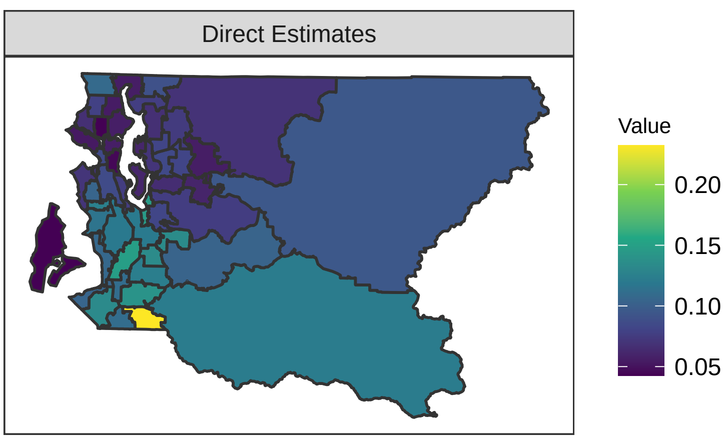
We map the posterior mean estimates.

```
data(KingCounty)
toplot <- smoothed$smooth
mapPlot(data = toplot, geo = KingCounty, variables = c("mean"),
    labels = c("Posterior Mean"), by.data = "region", by.geo = "HRA2010v2_")
```



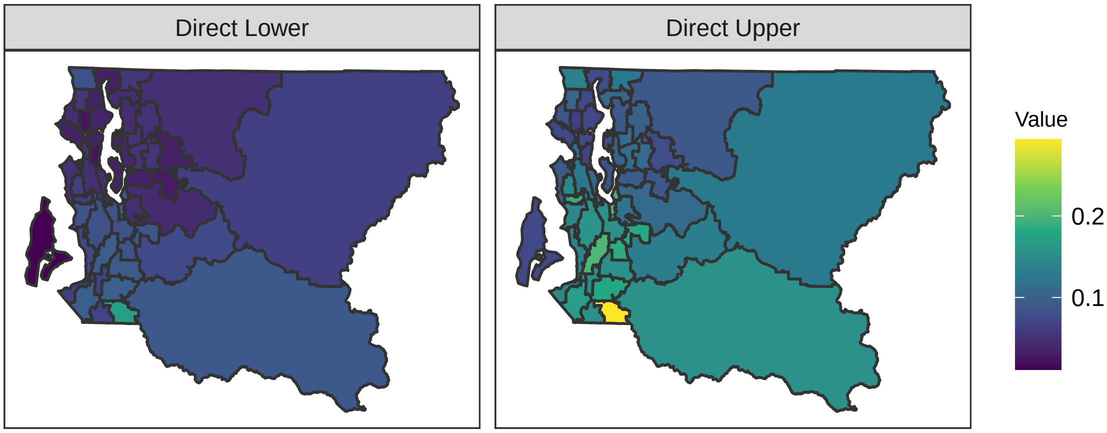
We map the Direct Estimates, which are available in the `smoothSurvey` fit.

```
toplot$HTest <- smoothed$HT$HT.est
mapPlot(data = toplot, geo = KingCounty, variables = c("HTest"),
    labels = c("Direct Estimates"), by.data = "region",
    by.geo = "HRA2010v2_")
```



Now map the lower and upper endpoints of 95% CI for direct estimates

```
lo <- smoothed$HT$HT.est - 1.96 * sqrt(smoothed$HT$HT.var)
hi <- smoothed$HT$HT.est + 1.96 * sqrt(smoothed$HT$HT.var)
toplot$HTlower <- lo
toplot$HTupper <- hi
mapPlot(data = toplot, geo = KingCounty, variables = c("HTlower",
    "HTupper"), labels = c("Direct Lower", "Direct Upper"), by.data = "region",
    by.geo = "HRA2010v2_")
```



Fit Fay-Herriot smoothing model, which acknowledges the design

We now acknowledge the design and fit the model

$$\widehat{\theta}_i \sim \mathrm{N}(\theta_i, \widehat{V}_i)$$

with $\widehat{\theta}_i = \log[\widehat{p}_i/(1-\widehat{p}_i)]$ where $\widehat{p}_i$ being the direct estimate and $\widehat{V}_i$ the variance of this estimate (where the design is acknowledged in the variance calculation) and

$$\theta_i = \log\left(\frac{p_i}{1-p_i}\right) = \mu + \epsilon_i$$

with $\epsilon_i \sim_{iid} N(0, \sigma^2)$.

We put `Amat=NULL` to obtain an iid model only (i.e., the standard Fay-Herriot model without covariates).

```
FHmodel <- smoothSurvey(data = BRFSS, geo = KingCounty, Amat = NULL,
    responseType = "binary", responseVar = "diab2", strataVar = "strata",
    weightVar = "rwt_llcp", regionVar = "hracode", clusterVar = "~1",
    CI = 0.95)
FHmodel$fit$summary.fixed[1:5]
##                  mean         sd 0.025quant  0.5quant 0.975quant
## (Intercept) -2.666314 0.07087866  -2.806351 -2.666159  -2.527137
FHmodel$fit$summary.hyper[1:5]
##                                 mean       sd 0.025quant 0.5quant 0.975quant
## Precision for region.struct 7.334463 2.562294   3.731434 6.881683   13.54568
sqrt(1/FHmodel$fit$summary.hyper[3:5])
##                             0.025quant  0.5quant 0.975quant
## Precision for region.struct  0.5176808 0.3811998  0.2717062
```

Now extend the random effects structure to allow for BYM2 random effects.

```
svysmoothed <- smoothSurvey(data = BRFSS, geo = KingCounty, Amat = mat,
    responseType = "binary", responseVar = "diab2", strataVar = "strata",
    weightVar = "rwt_llcp", regionVar = "hracode", clusterVar = "~1",
    CI = 0.95)
svysmoothed$fit$summary.fixed[1:5]
##                  mean         sd 0.025quant  0.5quant 0.975quant
## (Intercept) -2.669643 0.04711008   -2.76225 -2.669661  -2.576929
svysmoothed$fit$summary.hyper[1:2, 1:5]
##                                   mean        sd 0.025quant   0.5quant
## Precision for region.struct 11.4183553 4.2566023  5.3501092 10.6661839
```

```
## Phi for region.struct        0.8036819 0.1676342  0.3823004  0.8519589
##                             0.975quant
## Precision for region.struct 21.8586398
## Phi for region.struct        0.9917089
sqrt(1/svysmoothed$fit$summary.hyper[1, 3:5])
##                             0.025quant  0.5quant 0.975quant
## Precision for region.struct  0.4323333 0.3061931   0.213889
```

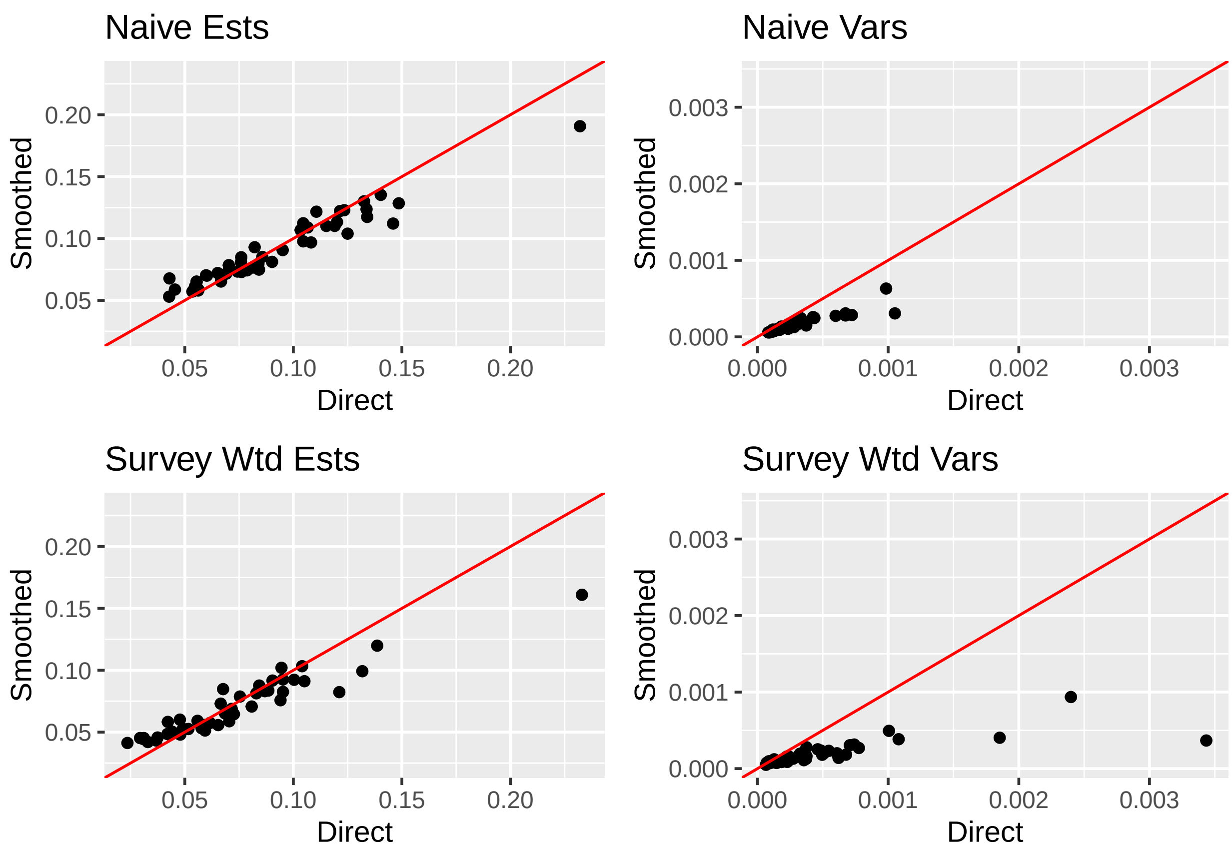
Now we can compile the four sets of estimates which either do or don't take into account the survey weights, and which either do or don't include smoothing over space. Then, create scatter plots to compare them.

```
est <- data.frame(
  naive = smoothed$HT$HT.est,
  weighted = svysmoothed$HT$HT.est,
  smooth = smoothed$smooth$mean,
  weightedsmooth = svysmoothed$smooth$mean
)
var <- data.frame(
  naive = smoothed$HT$HT.var,
  weighted = svysmoothed$HT$HT.var,
  smooth = smoothed$smooth$var,
  weightedsmooth = svysmoothed$smooth$var
)
```

```
l1 <- range(est)
l2 <- range(var)
library(ggplot2)
g1 <- ggplot(est, aes(x = naive, y = smooth)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  ggtitle("Naive Ests") +
  xlab("Direct") +
  ylab("Smoothed") +
  xlim(l1) +
  ylim(l1)
g2 <- ggplot(var, aes(x = naive, y = smooth)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  ggtitle("Naive Vars") +
  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 Wtd Ests") +
  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") +
```

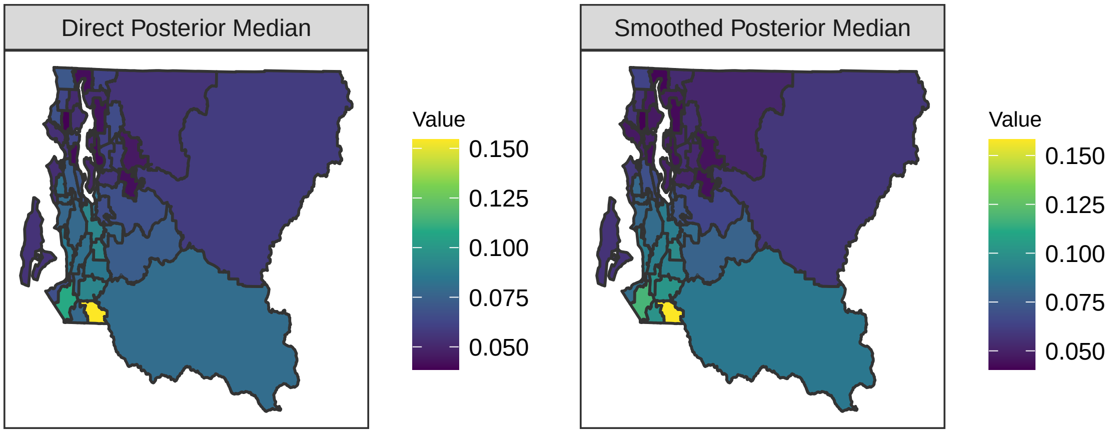
```
  ggtitle("Survey Wtd Vars") +
  xlab("Direct") +
  ylab("Smoothed") +
  xlim(l2) +
  ylim(l2)
```

```
library(gridExtra)
grid.arrange(grobs = list(g1, g2, g3, g4), ncol = 2)
```



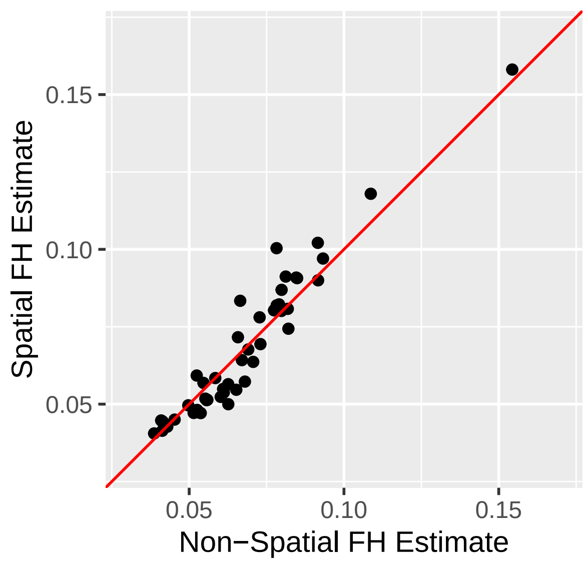
We can also compare posterior medians of the two Bayes Fay Harriot models (`FHmodel` and `svysmoothed`).

```
p1 <- mapPlot(data = FHmodel$smooth, geo = KingCounty, variables = "median",
    labels = "Direct Posterior Median", by.data = "region", by.geo = "HRA2010v2_")
p2 <- mapPlot(data = svysmoothed$smooth, geo = KingCounty, variables = "median",
    labels = "Smoothed Posterior Median", by.data = "region",
    by.geo = "HRA2010v2_")
grid.arrange(grobs = list(p1, p2), ncol = 2)
```



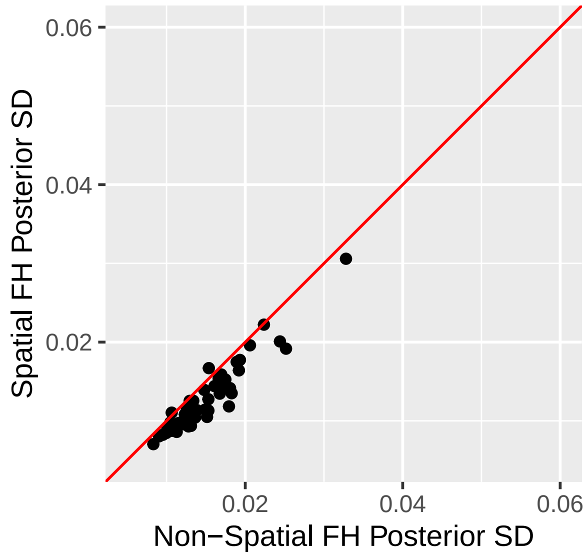
It's also useful to plot the estimates against each other and the posterior standard errors against each other.

```
# Posterior Estimates
smoothed_nonspatial <- FHmodel$smooth[, c("region", "median")]
smoothed_spatial <- svysmoothed$smooth[, c("region", "median")]
smoothed_df <- merge(smoothed_nonspatial, smoothed_spatial, by = "region")
names(smoothed_df) <- c("region", "nonspatial", "spatial")

ggplot(smoothed_df, aes(x = nonspatial, y = spatial)) + geom_point() +
    labs(y = "Spatial FH Estimate", x = "Non-Spatial FH Estimate") +
    geom_abline(color = "red") + coord_equal(xlim = c(0.03, 0.17),
    ylim = c(0.03, 0.17))
```




```
# Posterior SDs
smoothed_nonspatial_sd <- FHmodel$smooth[, c("region", "var")]
smoothed_spatial_sd <- svysmoothed$smooth[, c("region", "var")]
smoothed_sd_df <- merge(smoothed_nonspatial_sd, smoothed_spatial_sd,
    by = "region")
names(smoothed_sd_df) <- c("region", "nonspatial", "spatial")
smoothed_sd_df$spatial <- sqrt(smoothed_sd_df$spatial)  # convert variance to sd
smoothed_sd_df$nonspatial <- sqrt(smoothed_sd_df$nonspatial)

ggplot(smoothed_sd_df, aes(x = nonspatial, y = spatial)) + geom_point() +
```

```
    labs(y = "Spatial FH Posterior SD", x = "Non-Spatial FH Posterior SD") +
    geom_abline(color = "red") + coord_equal(xlim = c(0.005,
    0.06), ylim = c(0.005, 0.06))
```



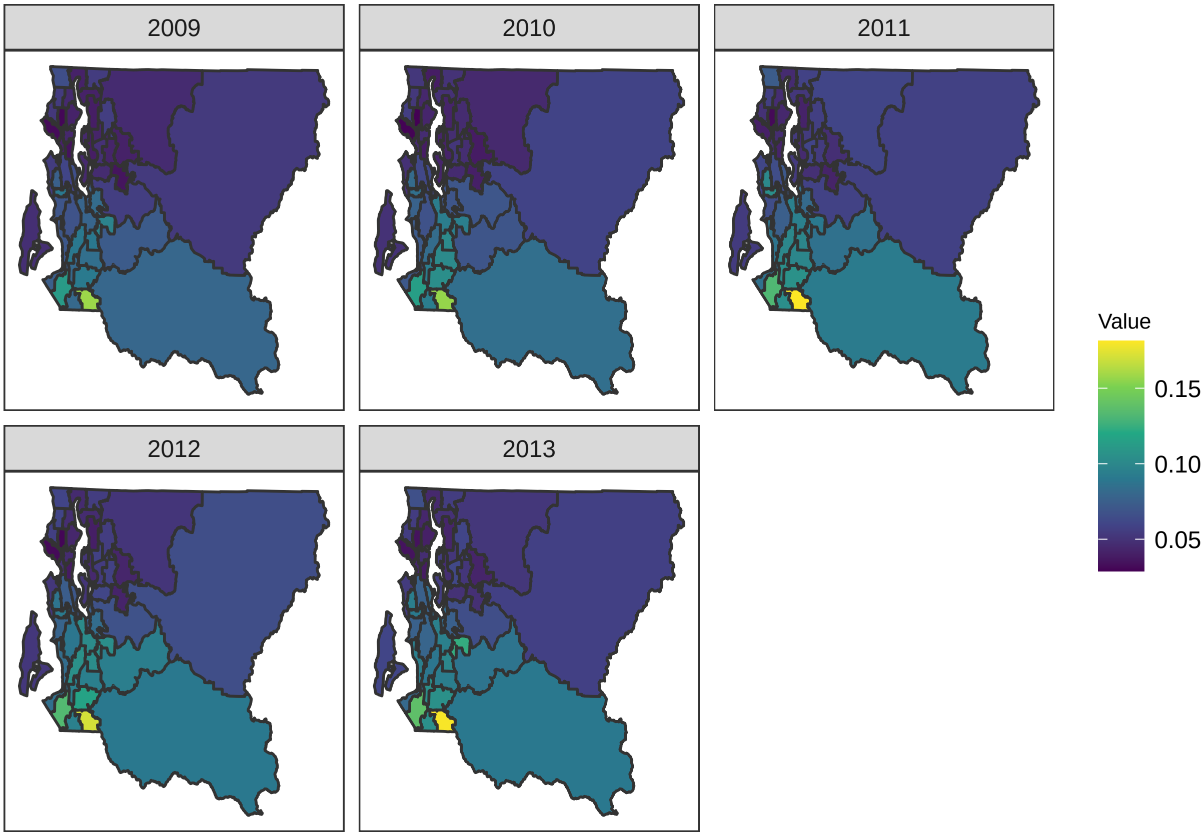
SAE in Space and Time

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 – see Held (2000, Statistics in Medicine).

```
svysmoothed.year <- fitGeneric(data = BRFSS, geo = KingCounty,
    Amat = mat, responseType = "binary", responseVar = "diab2",
    strataVar = "strata", weightVar = "rwt_llcp", regionVar = "hracode",
    clusterVar = "~1", timeVar = "year", time.model = "rw1",
    type.st = 1)
```

Maps of Posterior Means over Time

```
mapPlot(data = svysmoothed.year$smooth, geo = KingCounty, values = "mean",
    variables = "time", by.data = "region", by.geo = "HRA2010v2_",
    is.long = TRUE)
```



Final Comments

More materials can be found here: http://faculty.washington.edu/jonno/index.html.

`SUMMER` has a Github page with the latest changes, see also this paper:

Li ZR, Martin BD, Dong TQ, Fuglstad GA, Paige J, Riebler A, Clark S, Wakefield J. Space-time smoothing of demographic and health indicators using the R package SUMMER. arXiv preprint arXiv:2007.05117. 2020 Jul 10.