2022 SISCER SAE: Area-Level Models

Jon Wakefield and Peter Gao
Departments of Statistics and Biostatistics
University of Washington

2022-07-08

Notes Overview

In these notes we:

– Use the `sae` and `SUMMER` packages to fit various area-level models.

– We describe various approaches including direct estimation and Fay-Herriot modeling.

Direct Estimation

The direct (Horvitz-Thompson) estimator is:

$$\widehat{\overline{Y}}_i^{\text{DIR}} = \frac{1}{N_i}\sum_{k\in S_i} w_k y_k$$

- N_i is the population size of domain/area i
- S_i is the set of sampled observations in domain i
- w_k is the sampling weight for unit k, $k \in S_i$
- y_k is the observation for unit k, $k \in S_i$.

Post-stratified synthetic estimator

A post-stratified synthetic estimator is,

$$\widehat{\overline{Y}}_i^{\text{SYN}} = \frac{1}{N_i}\sum_{g=1}^{G} N_{ig}\widehat{\overline{y}}_{+g}^{\text{R}}$$

This estimator assumes that data are distributed into G post-strata that cut across the domains such that the within post-strata mean is constant across domains.

- N_{ig} is the size of the population in stratum g and domain d
- $\widehat{\overline{y}}_{+g}^{\text{R}} = \widehat{Y}_{+g}^{\text{DIR}}/\widehat{N}_{+g}^{\text{DIR}}$ is the (ratio) estimator of the stratum mean where $\widehat{y}_{+g}^{\text{DIR}}$ and $\widehat{N}_{+g}^{\text{DIR}}$ are HT estimators of the stratum total and stratum population size, respectively.

In the example we examine shortly, the strata correspond to education groups.

Poverty mapping

This example uses simulated data on income and other related variables to estimate incidence of poverty in Spanish counties.

Only sampling weights are provided.

Code/data from Molina et al. (2015).

```
library(sae)
data("incomedata")
data("sizeprov")
data("sizeprovedu")
```

Defining a poverty line

Molina et al (2015) define the poverty line `z`, and calculate an indicator with value 1 if the corresponding income value is below the poverty line and 0 otherwise.

```
povertyline <- 0.6 * median(incomedata$income)  # 6557.143
incomedata$poor <- as.integer(incomedata$income < povertyline)
```

Direct estimation with `sae`

We first obtain direct estimates using the `sea` package.

```
Popn <- sizeprov[, c("provlab", "Nd")]
sae.DIR <- direct(y = incomedata$poor, dom = incomedata$provlab,
    sweight = incomedata$weight, domsize = Popn)
head(sae.DIR)
##     Domain SampSize    Direct         SD       CV
## 1    Alava       96 0.2550373 0.04846645 19.00367
## 2 Albacete      173 0.1405924 0.03042195 21.63840
## 3 Alicante      539 0.2054832 0.02165788 10.53998
## 4  Almeria      198 0.2649583 0.04081541 15.40447
## 5    Avila       58 0.0551220 0.02555426 46.35946
## 6  Badajoz      494 0.2103349 0.02326525 11.06105
```

Direct estimation with `survey`

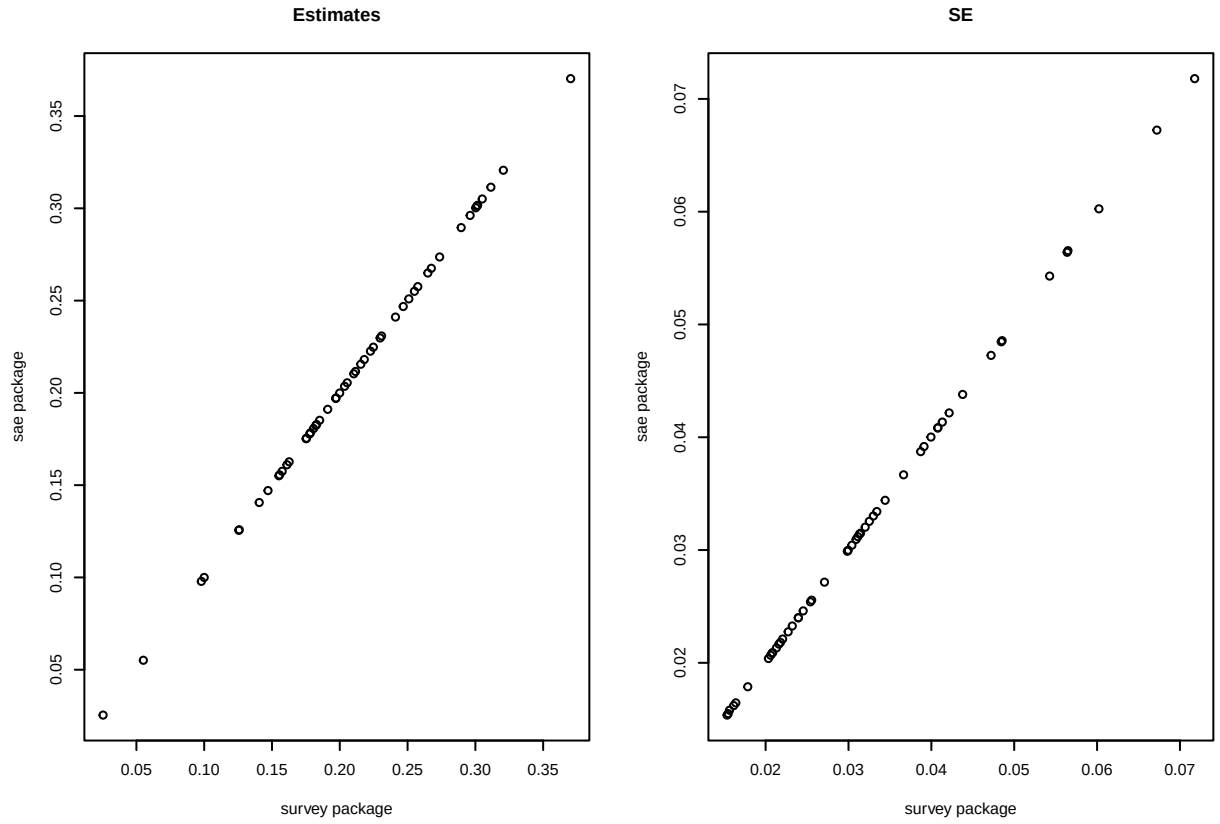
Now with the `survey` package.

```
library(survey)
incomedata$pop <- sum(sizeprov$Nd[match(incomedata$provlab, sizeprov$provlab)])
design <- svydesign(ids = ~1, weights = ~weight, data = incomedata,
    fpc = ~pop)
# Calulate HT estimates of totals
svy.DIR <- svyby(~poor, ~provlab, design, svytotal)
svy.DIR$prov_pop <- sizeprov$Nd[match(svy.DIR$provlab, sizeprov$provlab)]
# Calculate proportions from totals
svy.DIR$poor_mean = svy.DIR$poor/svy.DIR$prov_pop
svy.DIR$poor_mean_se = svy.DIR$se/svy.DIR$prov_pop
head(svy.DIR)
```

```
##           provlab       poor        se prov_pop poor_mean poor_mean_se
## Alava       Alava  75633.357 14364.585   296558 0.2550373   0.04843769
## Albacete Albacete  53623.777 11598.765   381413 0.1405924   0.03040999
## Alicante Alicante 352928.933 37106.979  1717556 0.2054832   0.02160452
## Almeria   Almeria 163783.945 25203.092   618150 0.2649583   0.04077181
## Avila       Avila   8992.713  4169.608   163142 0.0551220   0.02555815
## Badajoz   Badajoz 138863.753 15328.215   660203 0.2103349   0.02321743
```

Agreement between the two sets of estimates!

```
par(mfrow = c(1, 2), cex = 0.5)
plot(svy.DIR$poor_mean, sae.DIR$Direct, main = "Estimates", ylab = "sae package",
    xlab = "survey package")
plot(svy.DIR$poor_mean_se, sae.DIR$SD, main = "SE", ylab = "sae package",
    xlab = "survey package")
```



Post-stratified synthetic estimator

Post-stratified synthetic estimator using education: The `sizeprovedu` data frame provides population totals for different levels in each domain.

The `pssynt` function calculates stratum means (each level of education being a stratum) and combines to estimate overall domain means:

```
head(sizeprovedu)
##    provlab prov  educ0  educ1  educ2  educ3
```

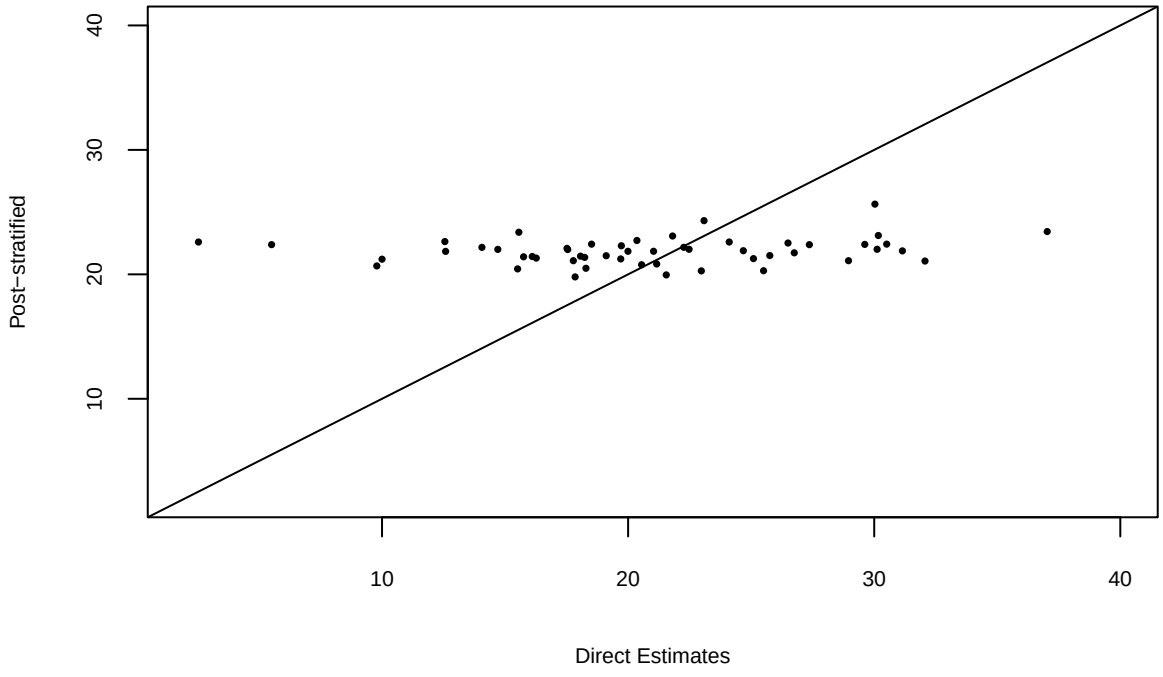
```
## 1    Alava    1  39486  68362 148028  40682
## 2 Albacete    2  63914 119907 156463  41129
## 3 Alicante    3 278004 393526 852714 193312
## 4  Almeria    4 111859 198535 248790  58966
## 5    Avila    5  20951  56992  67655  17544
## 6  Badajoz    6 112727 188708 293156  65612
Popn.educ <- sizeprovedu[, -2]
colnames(Popn.educ) <- c("provlab", "0", "1", "2", "3")
PSYN.educ <- pssynt(y = incomedata$poor, sweight = incomedata$weight,
    ps = incomedata$educ, domsizebyps = Popn.educ)
```

Results

Way too much shrinkage here to give seriously biased estimates.

```
results.DIR <- data.frame(Province = sae.DIR$Domain, SampleSize = sae.DIR$SampSize,
    DIR = sae.DIR$Direct * 100, PSYN.educ = PSYN.educ$PsSynthetic *
        100)
head(results.DIR, row.names = FALSE)
##   Province SampleSize      DIR PSYN.educ
## 1    Alava         96 25.50373  20.28781
## 2 Albacete        173 14.05924  22.16686
## 3 Alicante        539 20.54832  20.77174
## 4  Almeria        198 26.49583  22.51650
## 5    Avila         58  5.51220  22.38708
## 6  Badajoz        494 21.03349  21.85241
```

```
plot(results.DIR$DIR, results.DIR$PSYN.educ, xlab = "Direct Estimates",
    ylab = "Post-stratified", xlim = c(2, 40), ylim = c(2, 40),
    cex = 0.5, pch = 16, cex.lab = 0.7, cex.axis = 0.7)
abline(0, 1)
```



Area Level Fay-Herriot Models

Fay Herriot (1979) Model

First stage:

Let $\widehat{\theta}_i^{\text{DIR}}$ be a direct estimator of θ_i:

$$\widehat{\theta}_i^{\text{DIR}} = \theta_i + \epsilon_i; \qquad \epsilon_i \sim_{ind} N(0, V_i), \qquad i = 1, \ldots, n,$$

where V_i is the known sampling variance of the direct estimator $\widehat{\theta}_i^{\text{DIR}}$.

Second stage:

$$\theta_i = \mathbf{x}_i^T \boldsymbol{\beta} + \delta_i, \qquad \delta_i \sim_{iid} N(0, \sigma_\delta^2) \qquad i = 1, \ldots, n,$$

where σ_δ^2 is the between-area residual variance.

Combining the Two Stages:

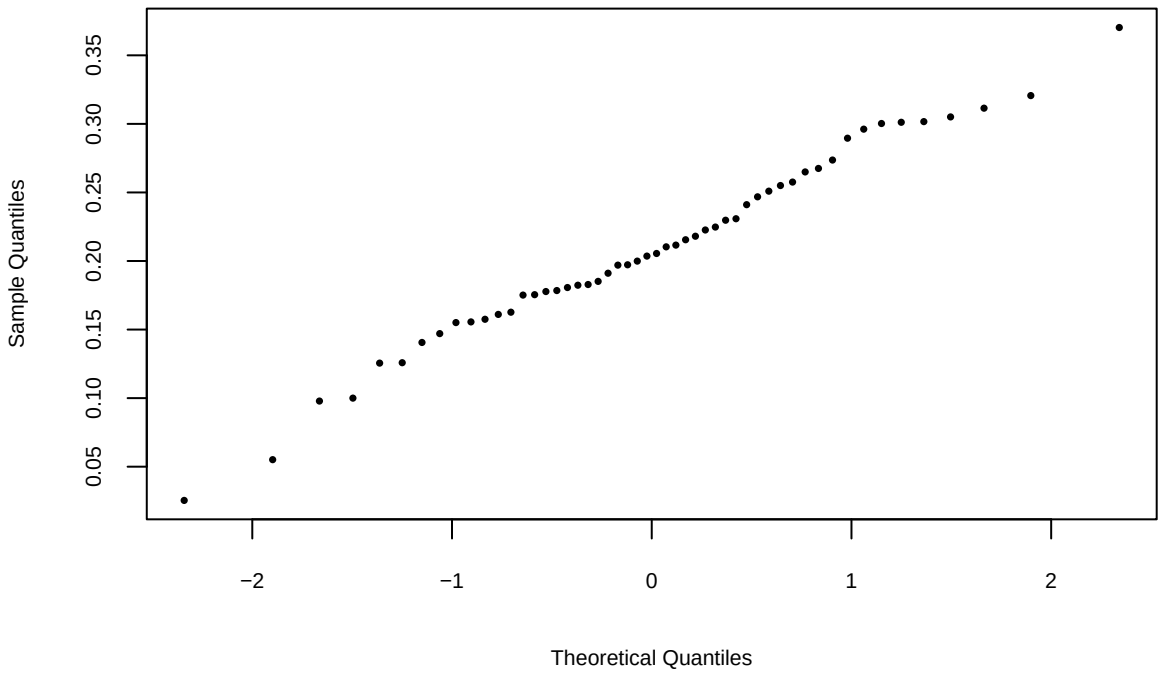
For $i = 1, \ldots, n$:

$$\theta_i^{\text{DIR}} = \mathbf{x}_i^T \boldsymbol{\beta} + \delta_i + \epsilon_i, \qquad \epsilon_i \sim_{ind} N(0, V_i), \qquad \delta_i \sim_{iid} N(0, \sigma_\delta^2)$$

We need to estimate β and σ_δ^2, along with the random effects δ_i.

Do the direct estimates look like a sample from a normal?

```
qqnorm(sae.DIR$Direct, main = "", cex = 0.5, pch = 16, cex.lab = 0.7,
    cex.axis = 0.7)
```



sae package

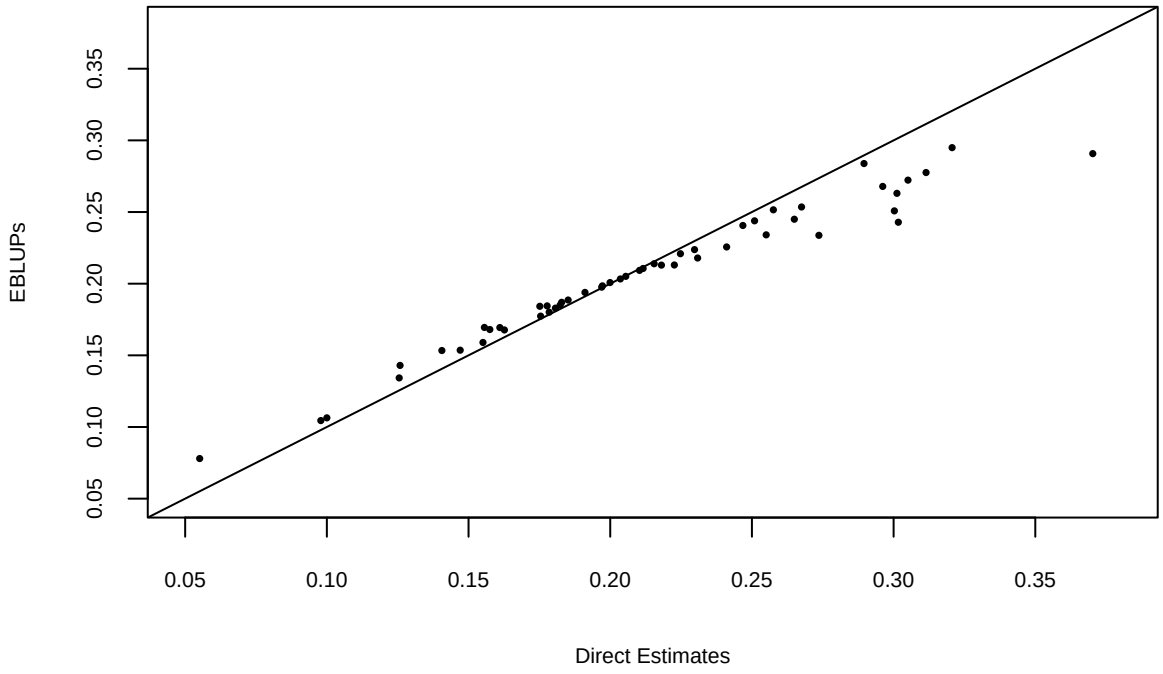
For this example there are no covariates so the two parameters to estimate are β_0 and σ_δ^2.

The results from the `sae` package are below.

```
income.FH <- mseFH(sae.DIR$Direct ~ 1, sae.DIR$SD^2)
income.FH$est$fit$estcoef
##        beta   std.error   tvalue        pvalue
## X 0.2023552 0.009449551 21.41426 9.838928e-102
income.FH$est$fit$refvar
## [1] 0.003554319
as.vector(income.FH$est$eblup)
##  [1] 0.23407448 0.15335213 0.20511853 0.24498009 0.07797403 0.20928033
##  [7] 0.10637332 0.28383394 0.20078095 0.25351502 0.15358331 0.16803768
## [13] 0.19846866 0.20329418 0.26786818 0.24386228 0.21792335 0.18449223
## [19] 0.27756427 0.18422431 0.22376988 0.14297048 0.22563144 0.27225562
## [25] 0.18859921 0.16947022 0.29076003 0.18010657 0.22091776 0.19390591
## [31] 0.17734507 0.15899465 0.21296200 0.25154358 0.24288263 0.16770579
## [37] 0.18513231 0.24060911 0.16938407 0.29493400 0.21304932 0.19758277
## [43] 0.05260739 0.26304154 0.18293284 0.23373402 0.13424405 0.21060694
## [49] 0.18699870 0.21390816 0.25079616 0.10448478
```

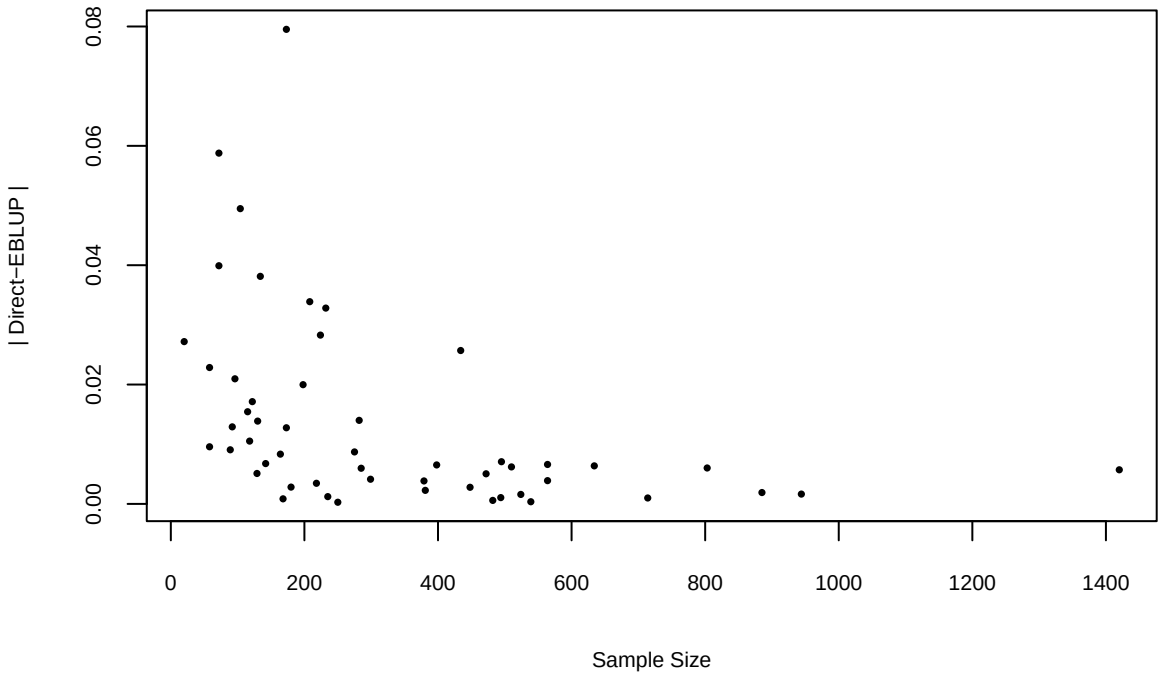
Shrinkage of F-H estimates

```
plot(income.FH$est$eblup ~ sae.DIR$Direct, xlim = c(0.05, 0.38),
    ylim = c(0.05, 0.38), xlab = "Direct Estimates", ylab = "EBLUPs",
    cex = 0.5, pch = 16, cex.lab = 0.7, cex.axis = 0.7)
abline(0, 1)
```



Shrinkage as a function of sample size

```
plot(abs(sae.DIR$Direct - income.FH$est$eblup) ~ sae.DIR$SampSiz,
    ylab = "| Direct-EBLUP |", xlab = "Sample Size", cex = 0.5,
    pch = 16, cex.lab = 0.7, cex.axis = 0.7)
```



Areas with larger sample sizes tend to have less shrinkage.

Milk expenditures

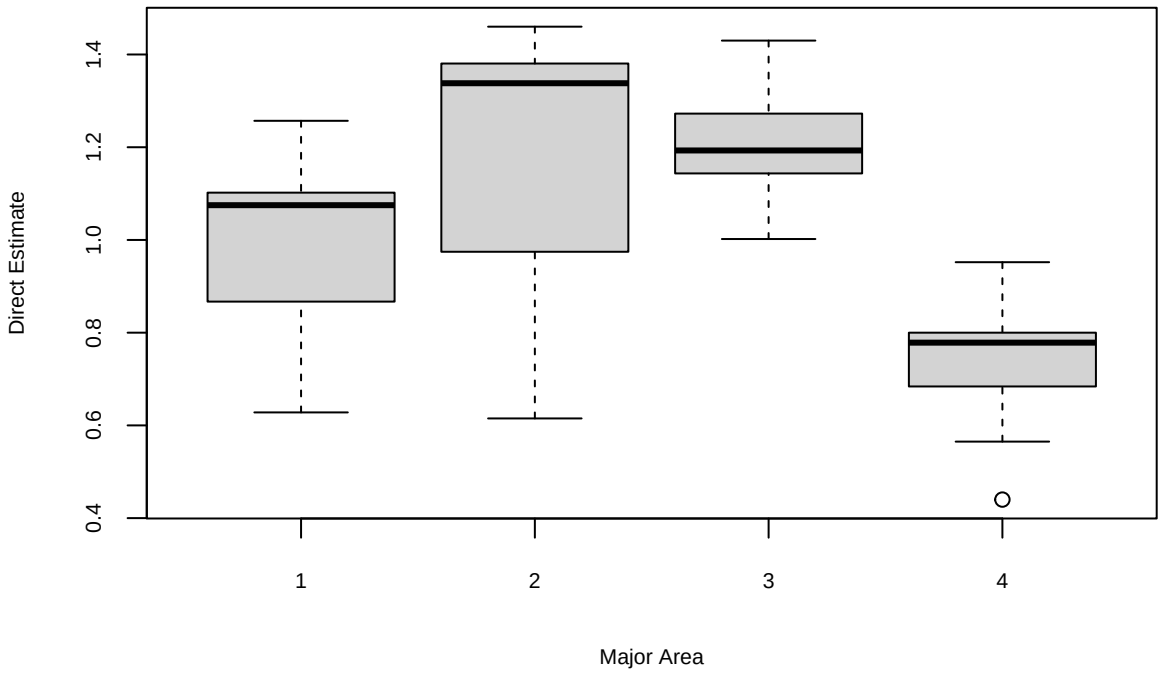
The `milk` dataset contains information on direct estimators for area level expenditures on milk based on data from the dairy survey component of the Consumer Expenditure Survey conducted by the U.S. Census Bureau. Weights are based on inverse sampling probabilities and adjusted for non-response/post-stratification. Code from Molina et al (2015). Data from Arora and Lahiri(1997).

```
library(sae)
data(milk)
head(milk)
##   SmallArea  ni    yi    SD    CV MajorArea
## 1         1 191 1.099 0.163 0.148         1
## 2         2 633 1.075 0.080 0.074         1
## 3         3 597 1.105 0.083 0.075         1
## 4         4 221 0.628 0.109 0.174         1
## 5         5 195 0.753 0.119 0.158         1
## 6         6 191 0.981 0.141 0.144         1
```

- `SmallArea`: areas of inferential interest.
- `ni`: area sample sizes.
- `yi`: average expenditure on fresh milk for the year 1989 in dollars (direct estimates).
- `SD`: estimated SD of direct estimators.
- `CV`: estimated coefficients of variation. of direct estimators
- `MajorArea`: major areas (US regions defined by You and Chapman, 2006).

Here we have a covariate, `MajorArea`.

```
par(mfrow = c(1, 1))
boxplot(milk$yi ~ milk$MajorArea, ylab = "Direct Estimate", xlab = "Major Area",
    cex.lab = 0.7, cex.axis = 0.7)
```



Fit the FH model

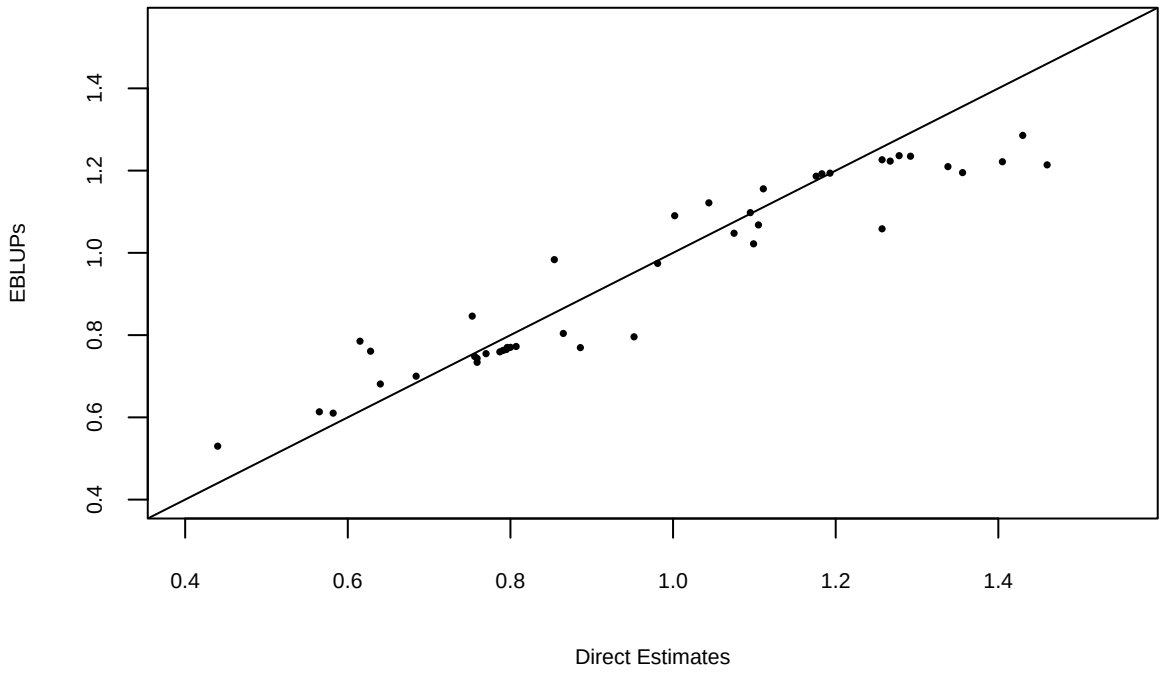
The `mseFH` function obtains EBLUPs under the Fay-Herriot model using REML as default for estimation of σ_δ^2.

Also estimates area specific mean squared errors (MSEs).

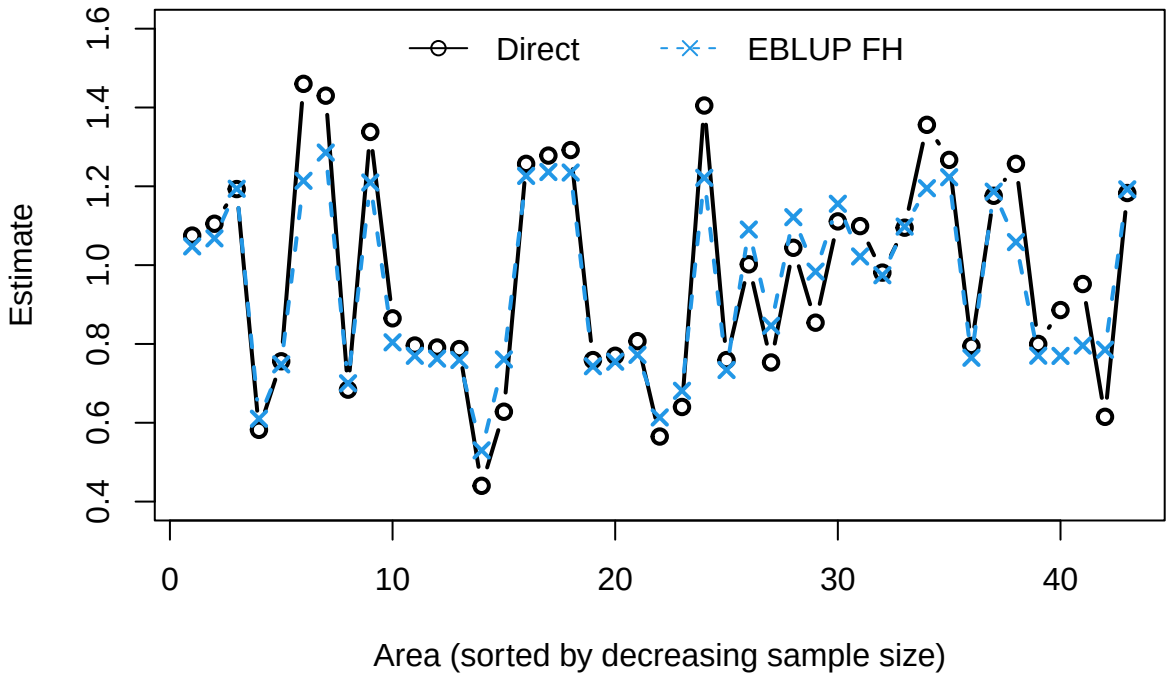
```
attach(milk)
FH <- mseFH(yi ~ as.factor(MajorArea), SD^2)
cv.FH <- 100 * sqrt(FH$mse)/FH$est$eblup
FH$est$fit$refvar  # random effects variance
## [1] 0.01855022
FH$est$fit$estcoef  # estimated fixed effects
##                              beta  std.error    tvalue       pvalue
## X(Intercept)            0.9681890 0.06936208 13.958476 2.793443e-44
## Xas.factor(MajorArea)2  0.1327801 0.10300072  1.289119 1.973569e-01
## Xas.factor(MajorArea)3  0.2269462 0.09232981  2.457995 1.397151e-02
## Xas.factor(MajorArea)4 -0.2413011 0.08161707 -2.956503 3.111496e-03
```

Shrinkage of F-H estimates

```
plot(FH$est$eblup ~ milk$yi, xlim = c(0.4, 1.55), ylim = c(0.4,
    1.55), xlab = "Direct Estimates", ylab = "EBLUPs", cex = 0.5,
    pch = 16, cex.lab = 0.7, cex.axis = 0.7)
abline(0, 1)
```



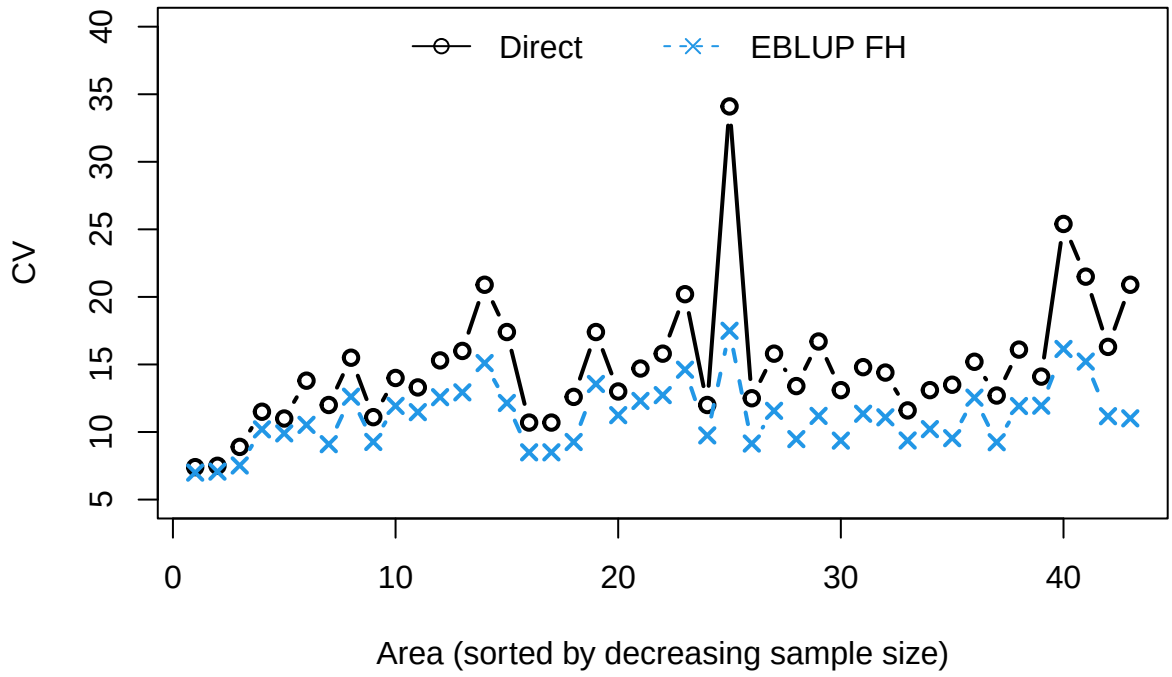

```
attach(milk)
results.FH <- data.frame(Area = SmallArea, SampleSize = ni, DIR = yi,
    cv.DIR = 100 * CV, eblup.FH = FH$est$eblup, cv.FH)
detach(milk)
results.FH <- results.FH[order(results.FH$SampleSize, decreasing = TRUE),
    ]
plot(results.FH$DIR, type = "n", ylab = "Estimate", ylim = c(0.4,
    1.6), xlab = "Area (sorted by decreasing sample size)")
points(results.FH$DIR, type = "b", col = 1, lwd = 2, pch = 1,
    lty = 1)
points(results.FH$eblup.FH, type = "b", col = 4, lwd = 2, pch = 4,
    lty = 2)
legend("top", legend = c("Direct", "EBLUP FH"), ncol = 2, col = c(1,
    4), bty = "n", pch = c(1, 4), lty = c(1, 2))
```



We see that points to the right (with the smallest sample size) undergo more shinrkage.

Coefficient of Variation

```
p2 <- plot(results.FH$cv.DIR, type = "n", ylab = "CV", ylim = c(5,
    40), xlab = "Area (sorted by decreasing sample size)")
points(results.FH$cv.DIR, type = "b", col = 1, lwd = 2, pch = 1,
    lty = 1)
points(results.FH$cv.FH, type = "b", col = 4, lwd = 2, pch = 4,
    lty = 2)
legend("top", legend = c("Direct", "EBLUP FH"), ncol = 2, col = c(1,
    4), bty = "n", pch = c(1, 4), lty = c(1, 2))
```



BRFSS 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 design weight.

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

```
library(SUMMER)
library(ggplot2)
library(patchwork)
data(BRFSS)
data(KingCounty)
BRFSS <- subset(BRFSS, !is.na(BRFSS$diab2))
BRFSS <- subset(BRFSS, !is.na(BRFSS$hracode))
```

Design object and direct estimates

We have stratified, disproportionate sampling, so note the arguments:

- `weights`
- `strata`

We then calculate the direct (weighted) estimates using the `survey` package.

```
design <- svydesign(ids = ~1, weights = ~rwt_llcp, strata = ~strata,
    data = BRFSS)
direct <- svyby(~diab2, ~hracode, design, svymean)
head(direct, n = 5)
```

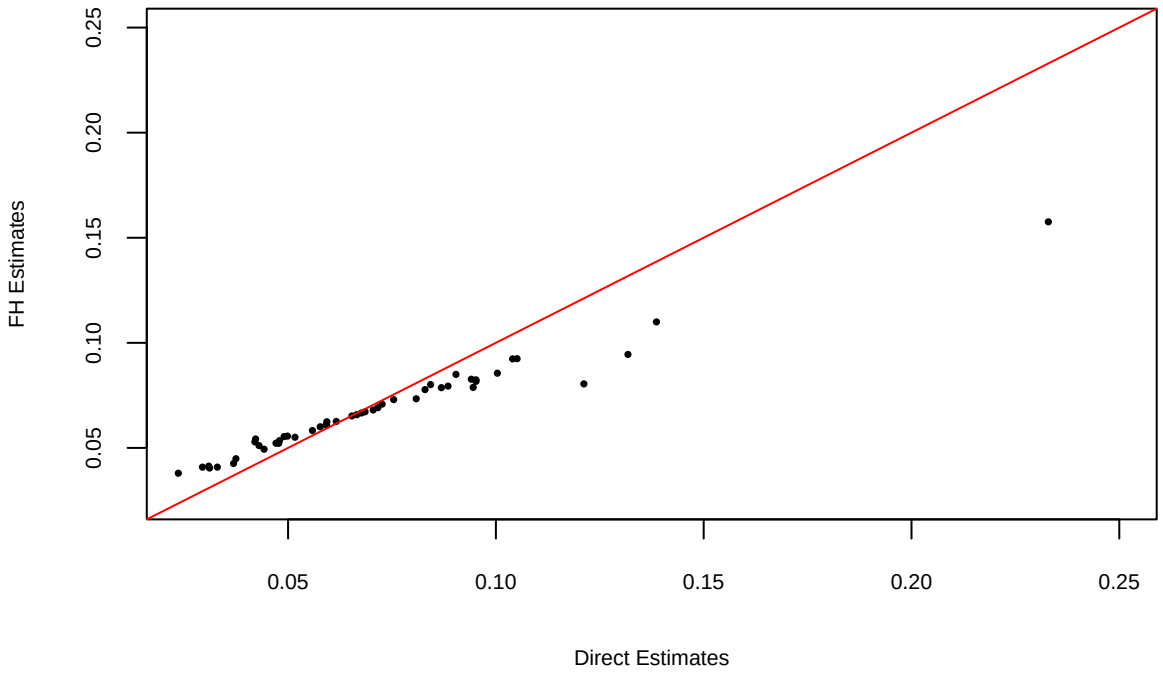
```
##                                                 hracode      diab2         se
## Auburn-North                                Auburn-North 0.10403154 0.02147752
## Auburn-South                                Auburn-South 0.23293289 0.04897800
## Ballard                                         Ballard 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
```

Results using sae

The `mseFH` model uses this to fit the FH model – we model the logits of the proportions which requires obtaining the variance of the direct estimates of the logits (using the delta mthod).

```
direct$var <- direct$se^2
direct$logit.diab2 <- logit(direct$diab2)
direct$logit.var <- direct$var/(direct$diab2^2 * (1 - direct$diab2)^2)
FH.brfss <- mseFH(logit.diab2 ~ 1, logit.var, data = direct)
FH.brfss$est$fit$estcoef
##        beta  std.error    tvalue pvalue
## X -2.666683 0.07089431 -37.61491      0
FH.brfss$est$fit$refvar
## [1] 0.1536226
```

```
plot(as.vector(expit(FH.brfss$est$eblup)) ~ direct$diab2, xlim = c(0.025,
    0.25), ylim = c(0.025, 0.25), xlab = "Direct Estimates",
    ylab = "FH Estimates", cex = 0.5, pch = 16, cex.lab = 0.7,
    cex.axis = 0.7)
abline(0, 1, col = "red")
```



Spatial Models: Implementing Fay-Herriot model with SUMMER

Income Example

We first fit the IID Fay-Herriot model - the same model as fit with the `sae` package, but now using Bayesian inference within `SUMMER`.

```
# need to give SUMMER an adjacency matrix, doesn't matter
# what
regions <- sae.DIR$Domain
Amat <- matrix(0, length(regions), length(regions))
colnames(Amat) <- rownames(Amat) <- regions
# fit the model
sae.DIR$Var <- sae.DIR$SD^2
summer.FH <- smoothSurvey(data = NULL, direct.est = sae.DIR,
    responseVar = "Direct", direct.est.var = "Var", regionVar = "Domain",
    Amat = NULL, responseType = "gaussian", pc.u = 0.5, pc.alpha = 0.05,
    CI = 0.95)
```

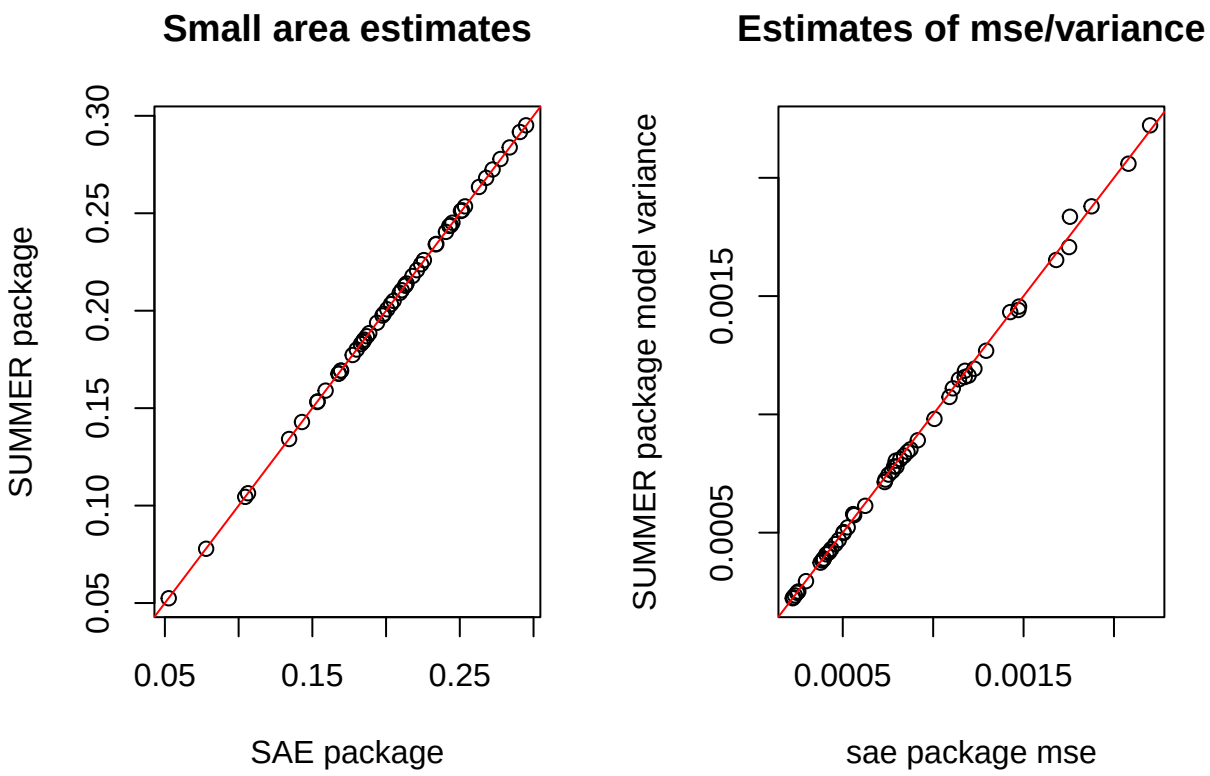
`SUMMER` results for income example

The fitted parameters from `sae` (obtained via likelihood-based methods) and estimated parameter posterior distribution from `SUMMER` (obtained from Bayesian methods, implemented via INLA) are in reasonable agreement. Below we show the estimated intercept β_0 and the between-area variance σ^2_δ (there are greater differences for the latter, but not as surprising since variances often do not symmetric posterior distributions).

```
income.FH$est$fit$estcoef  # sae intercept
##        beta   std.error   tvalue        pvalue
## X 0.2023552 0.009449551 21.41426 9.838928e-102
summer.FH$fit$summary.fixed[, c(1:5)]  # SUMMER intercept
##                  mean          sd 0.025quant  0.5quant 0.975quant
## (Intercept) 0.202426 0.009720188  0.1835659 0.2023282  0.2218369
income.FH$est$fit$refvar  # sae variance parameters
## [1] 0.003554319
1/summer.FH$fit$summary.hyperpar[, c(3)]  # SUMMER variance posterior
## [1] 0.006038998
```

Now compare EBLUP and hierarchical Bayes estimates:

```
par(mfrow = c(1, 2))
plot(income.FH$est$eblup, summer.FH$smooth$mean, xlab = "SAE package",
    ylab = "SUMMER package", main = "Small area estimates")
abline(0, 1, col = "red")
plot(income.FH$mse, summer.FH$smooth$var, xlab = "sae package mse",
    ylab = "SUMMER package model variance", main = "Estimates of mse/variance")
abline(0, 1, col = "red")
```



Milk Expenditure

Comparison with SUMMER

We fit the IID random effects only Fay-Herriot model using direct estimates.

Direct estimates as input requires two fields when calling `smoothSurvey()`

- Direct estimates (`HT.est` below).
- Variance of direct estimates (`HT.var` below).

We use the direct estimates from previous fit.

```
head(milk)
##   SmallArea  ni    yi    SD    CV MajorArea
## 1         1 191 1.099 0.163 0.148         1
## 2         2 633 1.075 0.080 0.074         1
## 3         3 597 1.105 0.083 0.075         1
## 4         4 221 0.628 0.109 0.174         1
## 5         5 195 0.753 0.119 0.158         1
## 6         6 191 0.981 0.141 0.144         1
milk$Var <- milk$SD^2
```

SUMMER Fit

```
# need to give SUMMER an adjacency matrix -- doesn't matter
# what
regions <- milk$SmallArea
```

```
Amat <- matrix(0, length(regions), length(regions))
colnames(Amat) <- rownames(Amat) <- regions
# Major Area fixed effects
Xmat <- milk[, c("SmallArea", "MajorArea")]
Xmat$MajorArea <- as.factor(Xmat$MajorArea)
# Fit the model with Major Area intercepts
summer.FH.milk <- smoothSurvey(data = NULL, direct.est = milk,
    X = Xmat, responseVar = "yi", direct.est.var = "Var", regionVar = "SmallArea",
    Amat = Amat, responseType = "gaussian", formula = "as.factor(MajorArea) + f(region.struct,
\t\tmodel = 'iid', hyper = hyperpc1)",
    pc.u = 1, pc.alpha = 0.01, CI = 0.95)
```

Lots of output when you fit a SUMMER model!

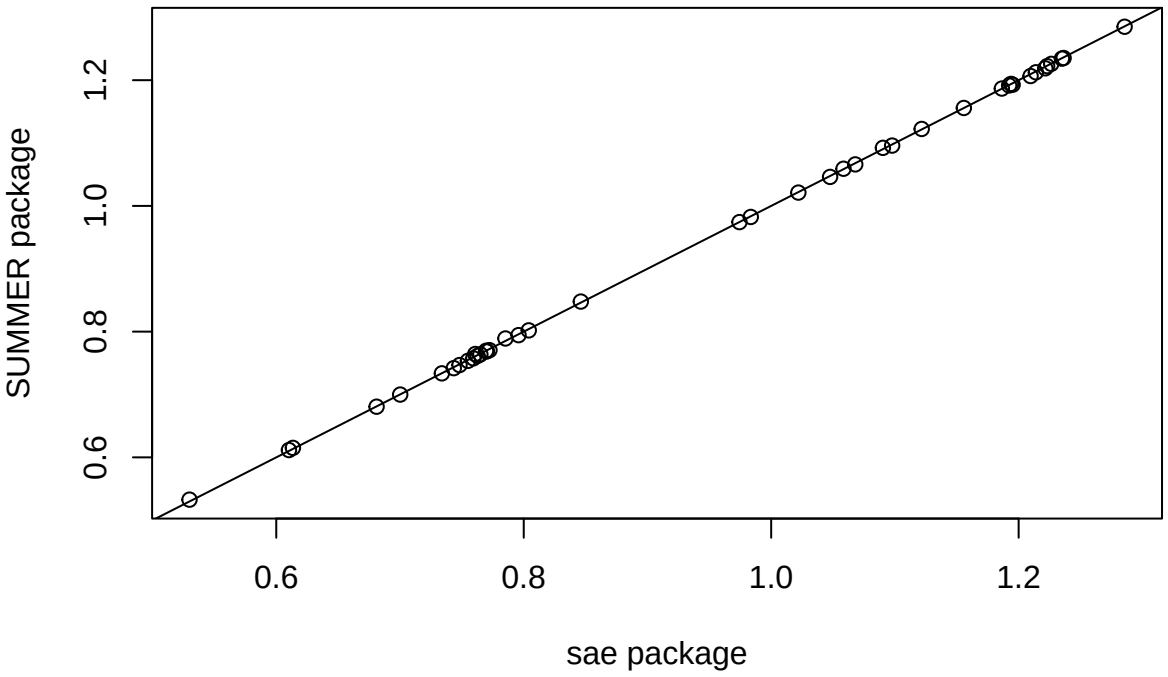
```
names(summer.FH.milk$fit)
##  [1] "names.fixed"                 "summary.fixed"
##  [3] "marginals.fixed"             "summary.lincomb"
##  [5] "marginals.lincomb"           "size.lincomb"
##  [7] "summary.lincomb.derived"     "marginals.lincomb.derived"
##  [9] "size.lincomb.derived"        "mlik"
## [11] "cpo"                         "gcpo"
## [13] "po"                          "waic"
## [15] "model.random"                "summary.random"
## [17] "marginals.random"            "size.random"
## [19] "summary.linear.predictor"    "marginals.linear.predictor"
## [21] "summary.fitted.values"       "marginals.fitted.values"
## [23] "size.linear.predictor"       "summary.hyperpar"
## [25] "marginals.hyperpar"          "internal.summary.hyperpar"
## [27] "internal.marginals.hyperpar" "offset.linear.predictor"
## [29] "model.spde2.blc"             "summary.spde2.blc"
## [31] "marginals.spde2.blc"         "size.spde2.blc"
## [33] "model.spde3.blc"             "summary.spde3.blc"
## [35] "marginals.spde3.blc"         "size.spde3.blc"
## [37] "logfile"                     "misc"
## [39] "dic"                         "mode"
## [41] "joint.hyper"                 "nhyper"
## [43] "version"                     "Q"
## [45] "graph"                       "ok"
## [47] "cpu.used"                    "all.hyper"
## [49] ".args"                       "call"
## [51] "model.matrix"
```

Comparison of FH estimates using sae and using SUMMER

```
FH$est$fit$refvar  # random effects variance
## [1] 0.01855022
summer.FH.milk$fit$summary.hyperpar[, c(1:5)]
##                                  mean       sd 0.025quant 0.5quant 0.975quant
## Precision for region.struct 61.32411 31.04339   25.38847 54.31158    138.255
```

Posterior median of σ_δ^2 is the reciprocal of the median for σ_δ^{-2}, which is 0.0184123.

```
plot(FH$est$eblup[as.numeric(summer.FH.milk$smooth$region)],
    summer.FH.milk$smooth$mean, xlab = "sae package", ylab = "SUMMER package")
abline(0, 1)
```



Grapes Example

SAR Fay-Herriot Model

The `sae` package also provides tools for implementing a spatial version of the Fay-Herriot model which assumes that the vector of area specific effects follows a first order simultaneous autoregressive, or SAR, process:

$$\delta = \rho_1 \mathbf{W}\delta + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(\mathbf{0}_n, \sigma_I^2 \mathbf{I}_n),$$

where $\mathbf{I}_n$ is the identity matrix for the n areas and $\mathbf{0}_n$ is a vector of zeroes of size D. Additionally, $\rho_1 \in (-1, 1)$ is an autoregression parameter and $\mathbf{W}$ is an adjacency matrix (with rows standardized to sum to 1).

The `mseSFH` function estimates the unknown variance parameters, the resulting EBLUP small area estimators, and then uses bootstrap methods to estimate the MSE of the estimators.

Grapes data

To illustrate the use of this function, Molina and Marhuenda (2015) consider a synthetic dataset concerning grape production surface area for 274 Italian municipalities. Below we load the relevant objects from the `sae` package. The `grapes` dataset containes direct estimators of the mean surface area in hectares for grape production in each municipality (`grapehect`), the sampling variance of these direct estimators (`var`), and relevant covariates including number of working dats and overall agrarian surface area. The `grapesprox` object contains the relevant adjacency matrix representing the municipalities' neighborhood structure.

```
data("grapes")
data("grapesprox")
```

Results using sae

```
sae.FH.grapes <- sae::mseSFH(grapehect ~ area + workdays - 1,
    var, grapesprox, data = grapes)
results <- data.frame(DIR = grapes$grapehect, eblup.SFH = sae.FH.grapes$est$eblup,
    mse = sae.FH.grapes$mse)
# reorder results for comparison later
results$area_name <- paste0("area_", rownames(results))
```

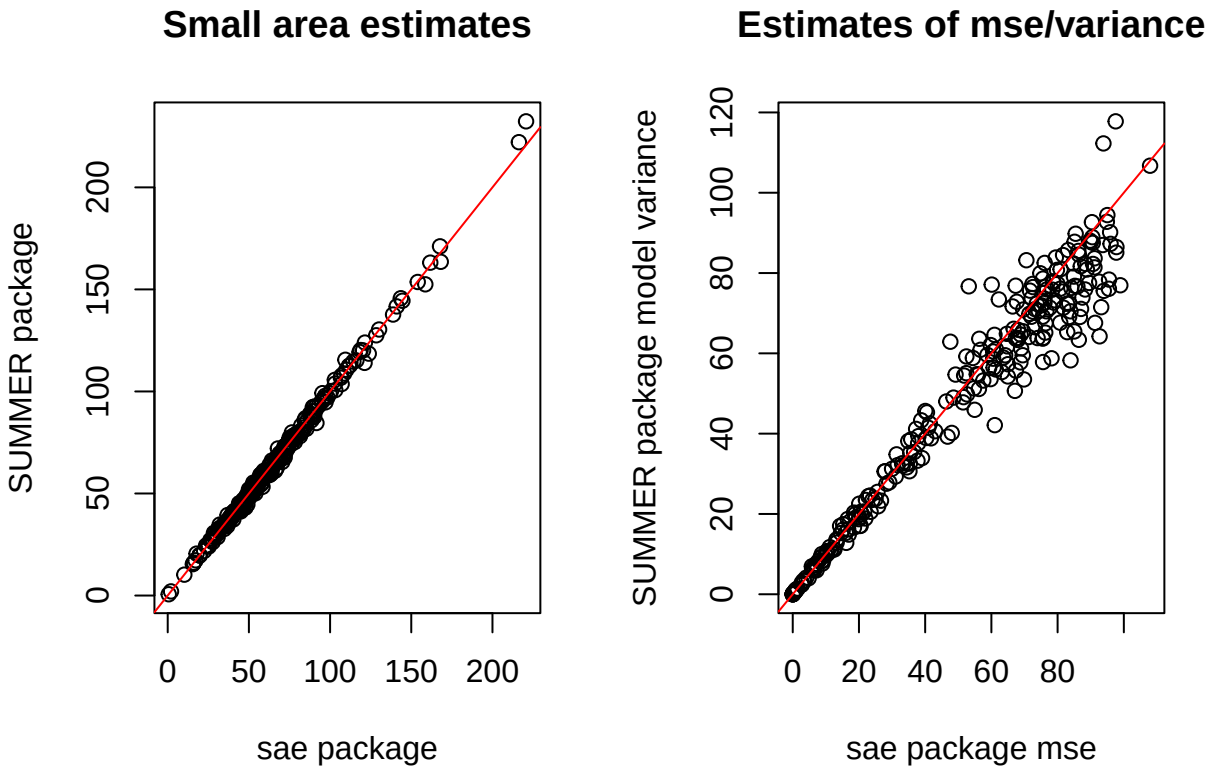
Results using SUMMER

The smoothSurvey function also allows the use of a model with spatially correlated area effects, but the default implementation assumes a BYM2 model for $[\delta_1, \ldots, \delta_n]$ rather than a simultaneous autoregressive model as in the SFH model implemented in sae.

```
# create area_name as SUMMER requires rownames of A_mat to
# match area variable
grapes$area_name <- paste0("area_", rownames(grapes))
Amat_grapes <- as.matrix(grapesprox)
rownames(Amat_grapes) <- colnames(Amat_grapes) <- grapes$area_name
X_grapes <- grapes[, c("area_name", "area", "workdays")]

# format direct estimates for SUMMER
grapes.dir <- grapes[, c(5, 1, 4)]
# scale direct estimates for use with INLA
grapes.dir$grapehect <- grapes.dir$grapehect/10
grapes.dir$var <- grapes.dir$var/100
summer.FH.grapes <- smoothArea(formula = grapehect ~ area + workdays,
    direct.est = grapes.dir, X.area = X_grapes, domain = ~area_name,
    Amat = Amat_grapes)
```

Despite the differing models, we again observe good agreement with the estimates, though less so with the estimates of uncertainty, which is interesting.

```
par(mfrow = c(1, 2))
plot(results$eblup.SFH, summer.FH.grapes$s.dir.sp.est$median *
    10, xlab = "sae package", ylab = "SUMMER package", main = "Small area estimates")
abline(0, 1, col = "red")
plot(results$mse, summer.FH.grapes$s.dir.sp.est$var * 100, xlab = "sae package mse",
    ylab = "SUMMER package model variance", main = "Estimates of mse/variance")
abline(0, 1, col = "red")
```



BRFSS 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 design weight.

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

```
library(SUMMER)
library(ggplot2)
library(patchwork)
data(BRFSS)
data(KingCounty)
BRFSS <- subset(BRFSS, !is.na(BRFSS$diab2))
BRFSS <- subset(BRFSS, !is.na(BRFSS$hracode))
```

Design object and direct estimates

We have stratified, disproportionate sampling, so note the arguments:

- `weights`
- `strata`

We then calculate the direct (weighted) esimates using the `survey` package.

```
library(survey)
design <- svydesign(ids = ~1, weights = ~rwt_llcp, strata = ~strata,
    data = BRFSS)
direct <- svyby(~diab2, ~hracode, design, svymean)
head(direct, n = 5)
##                                                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
```

In order to fit spatial smoothing models, we need the adjacency matrix for the HRAs, `mat` – here we use the `getAmap` function to extract from the map.

The `mseSFH` model uses this to fit the spatial SAR model.

```
mat <- getAmat(KingCounty, KingCounty$HRA2010v2_)
direct$var <- direct$se^2
direct$logit.diab2 <- SUMMER::logit(direct$diab2)
direct$logit.var <- direct$var/(direct$diab2^2 * (1 - direct$diab2)^2)
SFH.brfss <- sae::mseSFH(logit.diab2 ~ 1, logit.var, mat, data = direct)

results <- data.frame(region = direct$hracode, eblup.SFH = expit(SFH.brfss$est$eblup),
    mse = SFH.brfss$mse)
```

Results from the spatial Fay-Herriot fit

```
head(results, n = 5)
##                         region  eblup.SFH        mse
## 1                Auburn-North 0.09870261 0.03764429
## 2                Auburn-South 0.15881741 0.05163813
## 3                     Ballard 0.06665482 0.07018793
## 4          Beacon/Gtown/S.Park 0.06604920 0.05917442
## 5 Bear Creek/Carnation/Duvall 0.05717739 0.04162826
SFH.brfss$est$fit$estcoef
##        beta  std.error    tvalue pvalue
## X -2.655578 0.06387184 -41.57666      0
SFH.brfss$est$fit$refvar
## [1] 0.1742069
SFH.brfss$est$fit$spatialcorr
## [1] 0.4343474
```

In order to fit spatial smoothing models, we need the adjacency matrix for the HRAs, `mat` – here we use the `getAmap` function to extract from the map.

The `mseSFH` model uses this to fit the spatial SAR model.

```
mat <- getAmat(KingCounty, KingCounty$HRA2010v2_)
direct$var <- direct$se^2
direct$logit.diab2 <- SUMMER::logit(direct$diab2)
direct$logit.var <- direct$var/(direct$diab2^2 * (1 - direct$diab2)^2)
SFH.brfss <- sae::mseSFH(logit.diab2 ~ 1, logit.var, mat, data = direct)
results <- data.frame(region = direct$hracode, eblup.SFH = expit(SFH.brfss$est$eblup),
    mse = SFH.brfss$mse)
```

Results from the spatial Fay-Herriot fit

```
head(results, n = 5)
##                        region  eblup.SFH        mse
## 1               Auburn-North 0.09870261 0.03764429
## 2               Auburn-South 0.15881741 0.05163813
## 3                    Ballard 0.06665482 0.07018793
## 4         Beacon/Gtown/S.Park 0.06604920 0.05917442
## 5 Bear Creek/Carnation/Duvall 0.05717739 0.04162826
SFH.brfss$est$fit$estcoef
##        beta  std.error    tvalue pvalue
## X -2.655578 0.06387184 -41.57666      0
SFH.brfss$est$fit$refvar
## [1] 0.1742069
SFH.brfss$est$fit$spatialcorr
## [1] 0.4343474
```

Results using `SUMMER`

We now fit a hierarchical Bayes (Fay-Herriot) model using the `SUMMER` package.

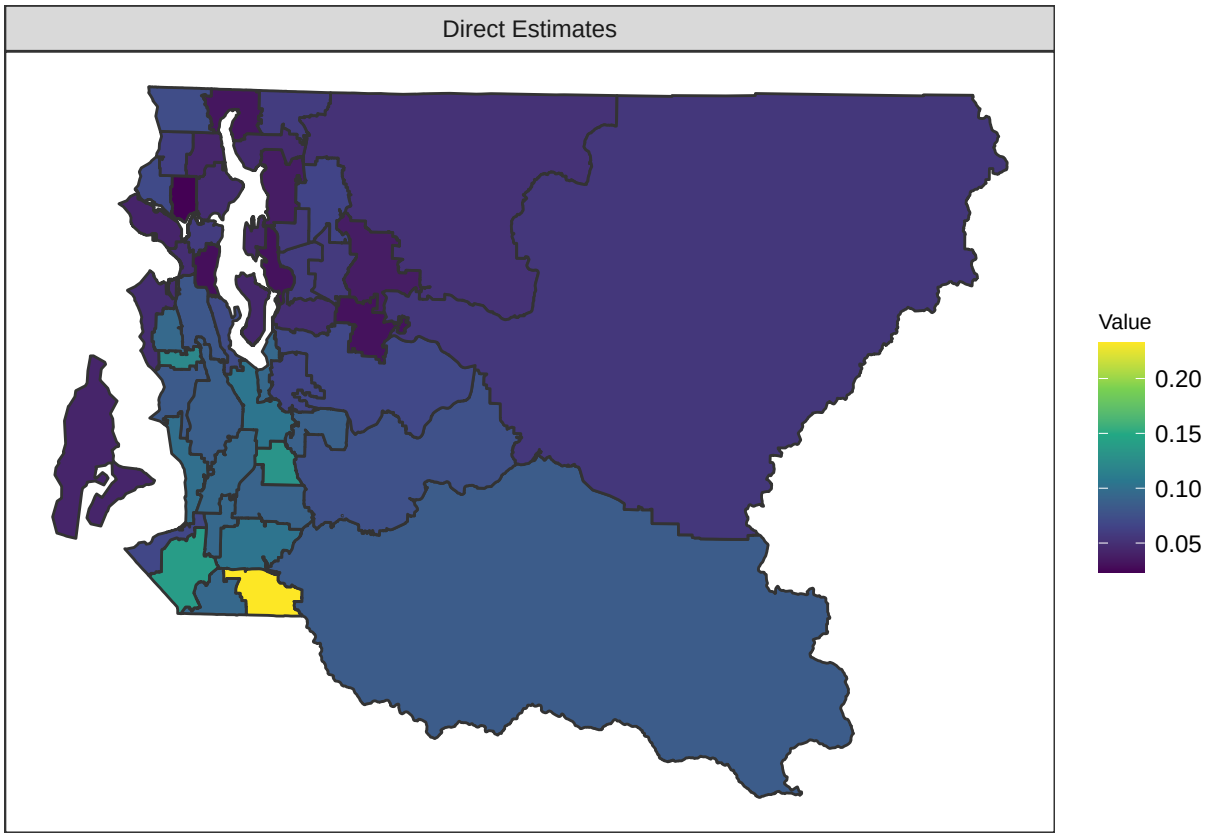
This uses a BYM2 spatial model.

```
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$smooth$mean
##  [1] 0.10262931 0.15944622 0.05820129 0.07009437 0.05231212 0.05574886
##  [7] 0.05236540 0.05213875 0.04586104 0.08784579 0.05446787 0.08133393
## [13] 0.05057383 0.04567151 0.07896039 0.08215285 0.09243458 0.05201250
## [19] 0.10312908 0.08344724 0.12236693 0.08620763 0.04190669 0.04480301
## [25] 0.04241019 0.09875389 0.09181260 0.09278794 0.04349197 0.04864848
## [31] 0.05036207 0.04758055 0.06469427 0.08157018 0.04794420 0.05718141
## [37] 0.04837067 0.05488169 0.07347741 0.07491669 0.09047394 0.04575839
## [43] 0.06846721 0.08301847 0.06414266 0.05900593 0.05836179 0.06066788
```

```
# Posterior summaries for intercept:
svysmoothed$fit$summary.fixed
##                    mean         sd 0.025quant  0.5quant 0.975quant mode
## (Intercept) -2.669588 0.04478383  -2.757548 -2.669604  -2.581533   NA
##                       kld
## (Intercept) 1.508409e-09
# Posterior summaries for between-area total variance:
1/svysmoothed$fit$summary.hyperpar[1, 3:5]
##                              0.025quant   0.5quant 0.975quant
## Precision for region.struct  0.1877719 0.09410157 0.04587125
# Posterior summaries for proportion of variance that is
# spatial:
svysmoothed$fit$summary.hyperpar[, 2]
## [1] 4.2480511 0.1709468
```

Map the Direct Estimates

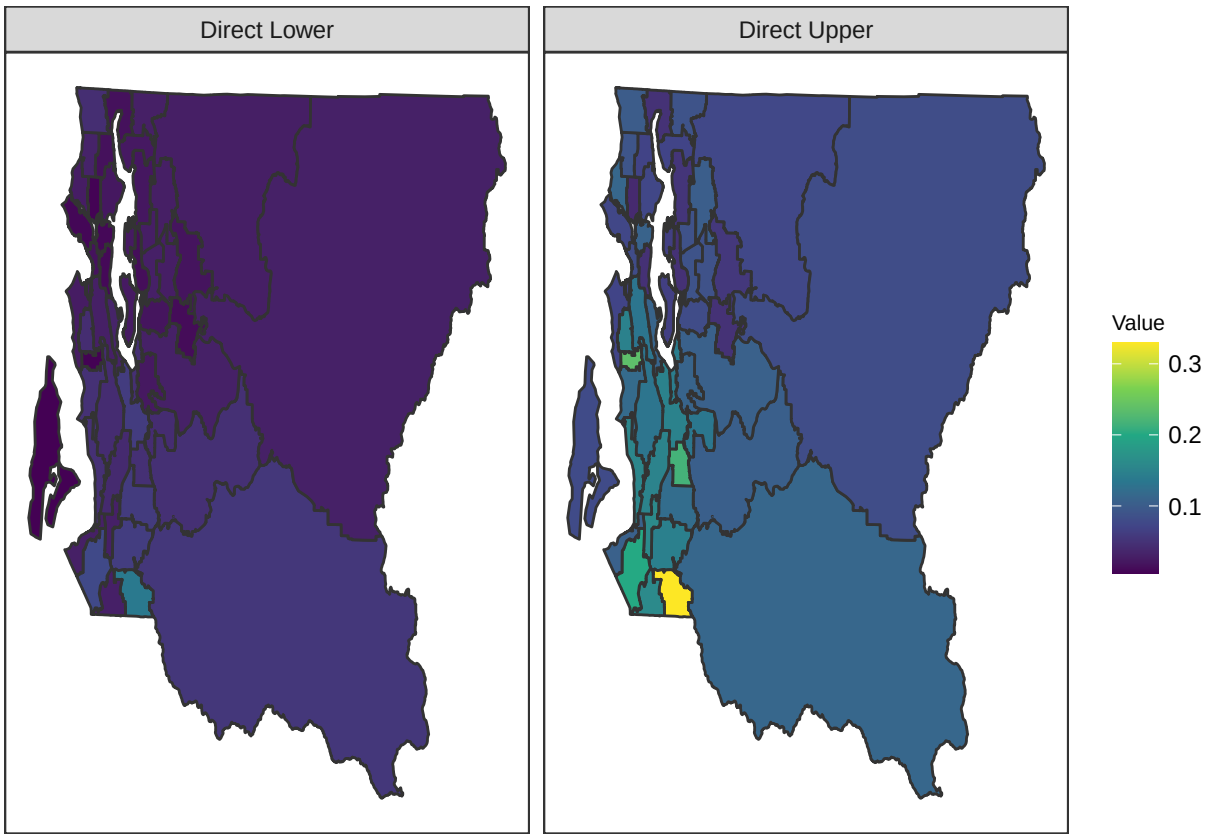
```
toplot0 <- svysmoothed$smooth
toplot0$HTest <- svysmoothed$HT$HT.est
mapPlot(data = toplot0, geo = KingCounty, variables = c("HTest"),
    labels = c("Direct Estimates"), by.data = "region", by.geo = "HRA2010v2_")
```



Map the Lower and Upper Endpoints of 95% CI for Direct Estimates

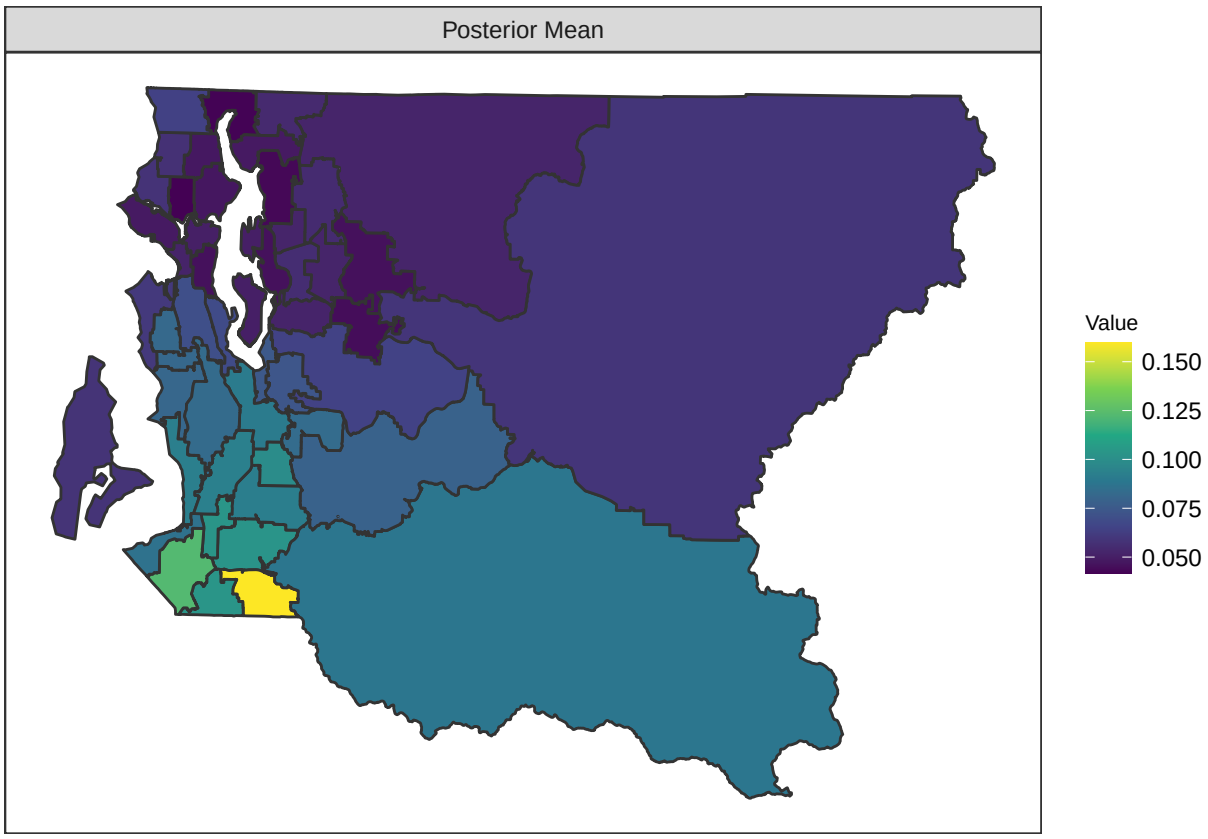
```
lo <- svysmoothed$HT$HT.est - 1.96 * sqrt(svysmoothed$HT$HT.var)
hi <- svysmoothed$HT$HT.est + 1.96 * sqrt(svysmoothed$HT$HT.var)
toplot0$HTlower <- lo
toplot0$HTupper <- hi
```

```
mapPlot(data = toplot0, geo = KingCounty, variables = c("HTlower",
    "HTupper"), labels = c("Direct Lower", "Direct Upper"), by.data = "region",
    by.geo = "HRA2010v2_")
```



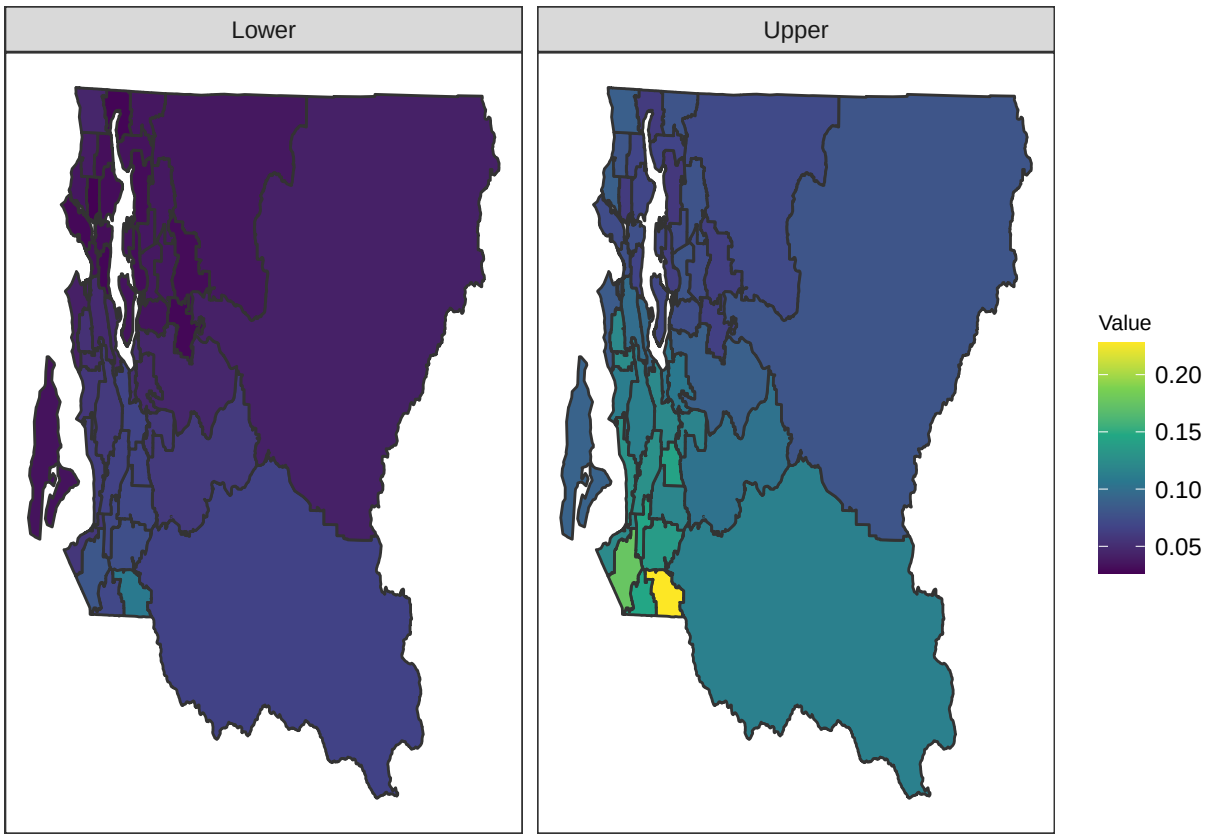
Map the Posterior Mean Estimates

```
mapPlot(data = toplot0, geo = KingCounty, variables = c("mean"),
    labels = c("Posterior Mean"), by.data = "region", by.geo = "HRA2010v2_")
```



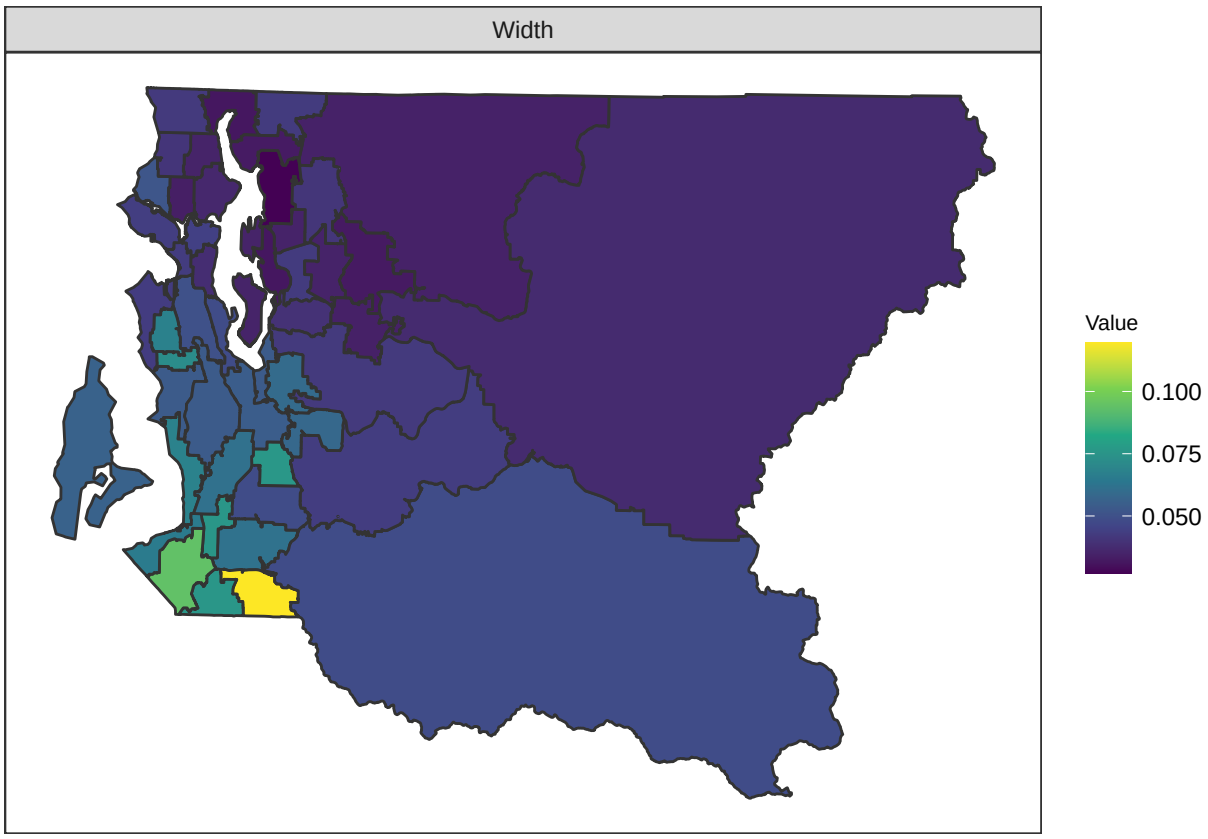
Map the 2.5% and 97.5% Posterior Quantiles

```
mapPlot(data = toplot0, geo = KingCounty, variables = c("lower",
    "upper"), labels = c("Lower", "Upper"), by.data = "region",
    by.geo = "HRA2010v2_")
```



Map the Interval Width

```
toplot0$width <- toplot0$upper - toplot0$lower
mapPlot(data = toplot0, geo = KingCounty, variables = c("width"),
    labels = c("Width"), by.data = "region", by.geo = "HRA2010v2_")
```

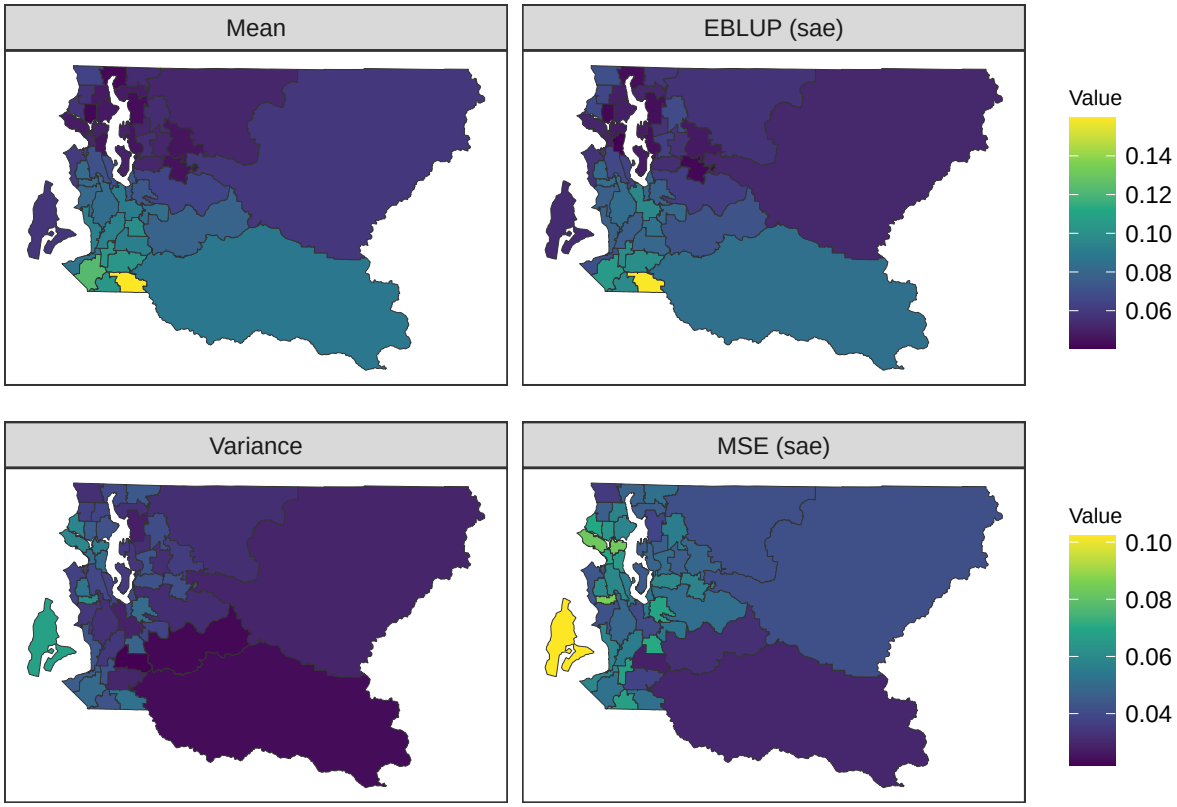


Comparison

We compare the `sae` and `SUMMER` spatially smoothed estimates.

The point estimates are the probability scale, while the uncertatiny measures are on the logistic scale.

```
toplot <- svysmoothed$smooth
toplot$logit.var <- toplot$var/(svysmoothed$smooth$mean^2 * (1 -
    svysmoothed$smooth$mean)^2)
toplot$mean.sae <- results$eblup.SFH
toplot$mse.sae <- results$mse
variables <- c("mean", "mean.sae", "logit.var", "mse.sae")
names <- c("Mean", "EBLUP (sae)", "Variance", "MSE (sae)")
g1 <- mapPlot(data = toplot, geo = KingCounty, variables = variables[1:2],
    labels = names[1:2], by.data = "region", by.geo = "HRA2010v2_",
    size = 0.1)
g2 <- mapPlot(data = toplot, geo = KingCounty, variables = variables[3:4],
    labels = names[3:4], by.data = "region", by.geo = "HRA2010v2_",
    size = 0.1)
g1/g2
```



Prior sensitivity

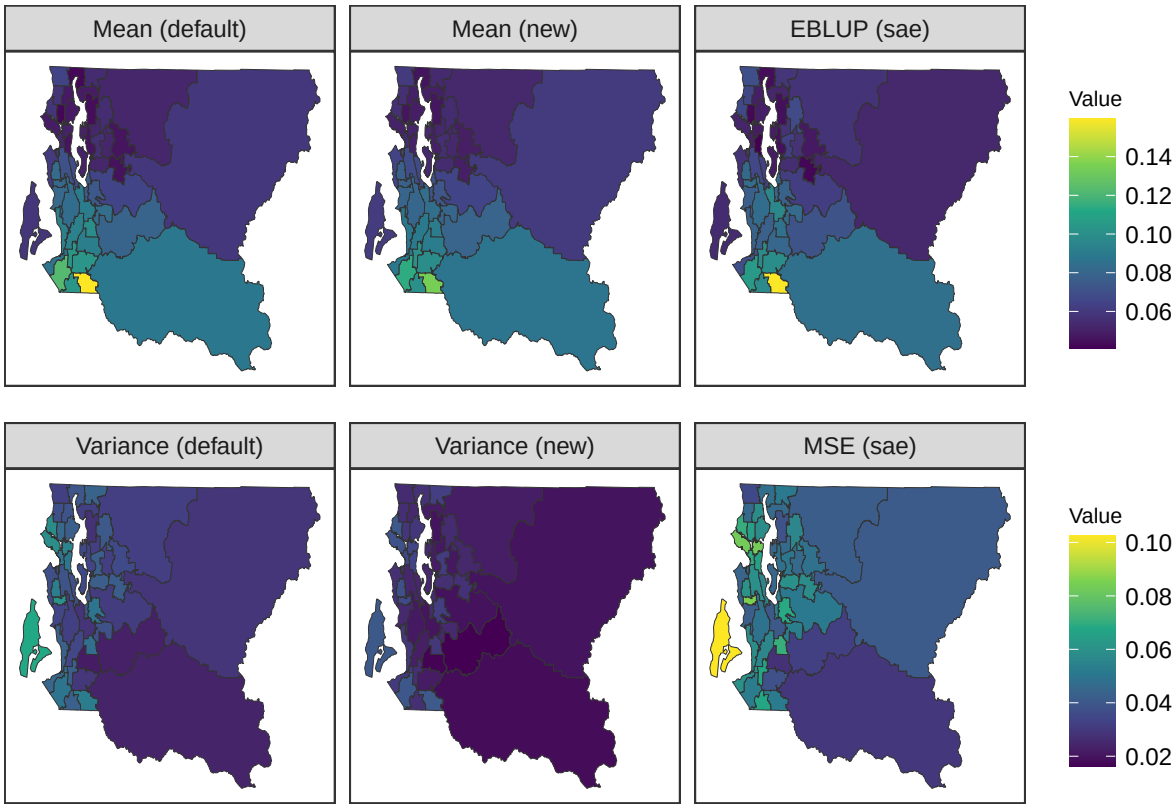
From the SUMMER vignette: If we change pc.u and pc.alpha from the default value $u = 1, \alpha = 0.01$ to $u = 0.1, \alpha = 0.01$, we would assign more prior mass on smaller variance of the random effects (to give more smoothing).

```
svysmoothed.1 <- smoothSurvey(data = BRFSS, geo = KingCounty,
    Amat = mat, responseType = "binary", responseVar = "diab2",
    strataVar = "strata", weightVar = "rwt_llcp", regionVar = "hracode",
    clusterVar = "~1", CI = 0.95, pc.u = 0.1, pc.alpha = 0.01)
toplot$mean.new <- svysmoothed.1$smooth$mean
toplot$logit.var.new <- svysmoothed.1$smooth$var/(svysmoothed.1$smooth$mean^2 *
    (1 - svysmoothed.1$smooth$mean)^2)
```

Map comparison

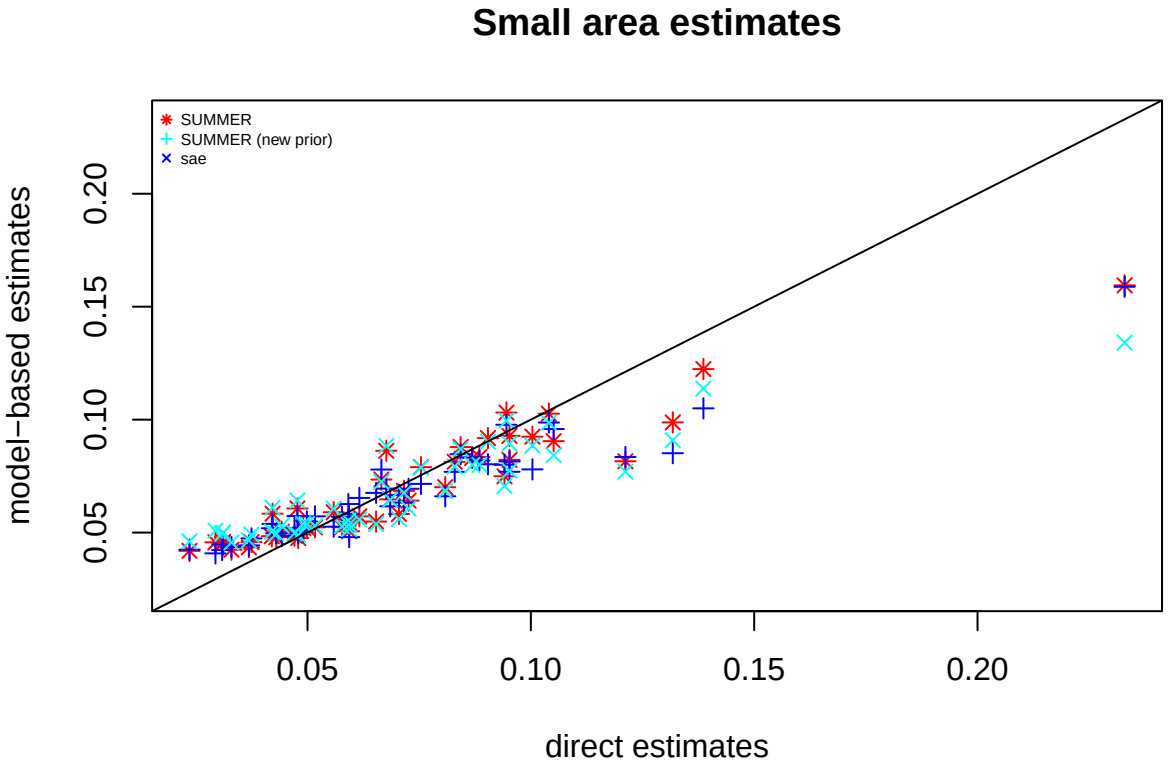
```
toplot2 <- svysmoothed$smooth
toplot2$logit.var <- toplot$var/(svysmoothed$smooth$mean^2 *
    (1 - svysmoothed$smooth$mean)^2)
toplot2$logit.var.new <- svysmoothed.1$smooth$var/(svysmoothed.1$smooth$mean^2 *
    (1 - svysmoothed.1$smooth$mean)^2)

toplot2$mean.new <- svysmoothed.1$smooth$mean
toplot2$mean.sae <- results$eblup.SFH
toplot2$mse.sae <- results$mse
variables <- c("mean", "mean.new", "mean.sae", "logit.var", "logit.var.new",
```

```
    "mse.sae")
names <- c("Mean (default)", "Mean (new)", "EBLUP (sae)", "Variance (default)",
    "Variance (new)", "MSE (sae)")
h1 <- mapPlot(data = toplot2, geo = KingCounty, variables = variables[1:3],
    labels = names[1:3], by.data = "region", by.geo = "HRA2010v2_",
    size = 0.1)
h2 <- mapPlot(data = toplot2, geo = KingCounty, variables = variables[4:6],
    labels = names[4:6], by.data = "region", by.geo = "HRA2010v2_",
    size = 0.1)
h1/h2
```



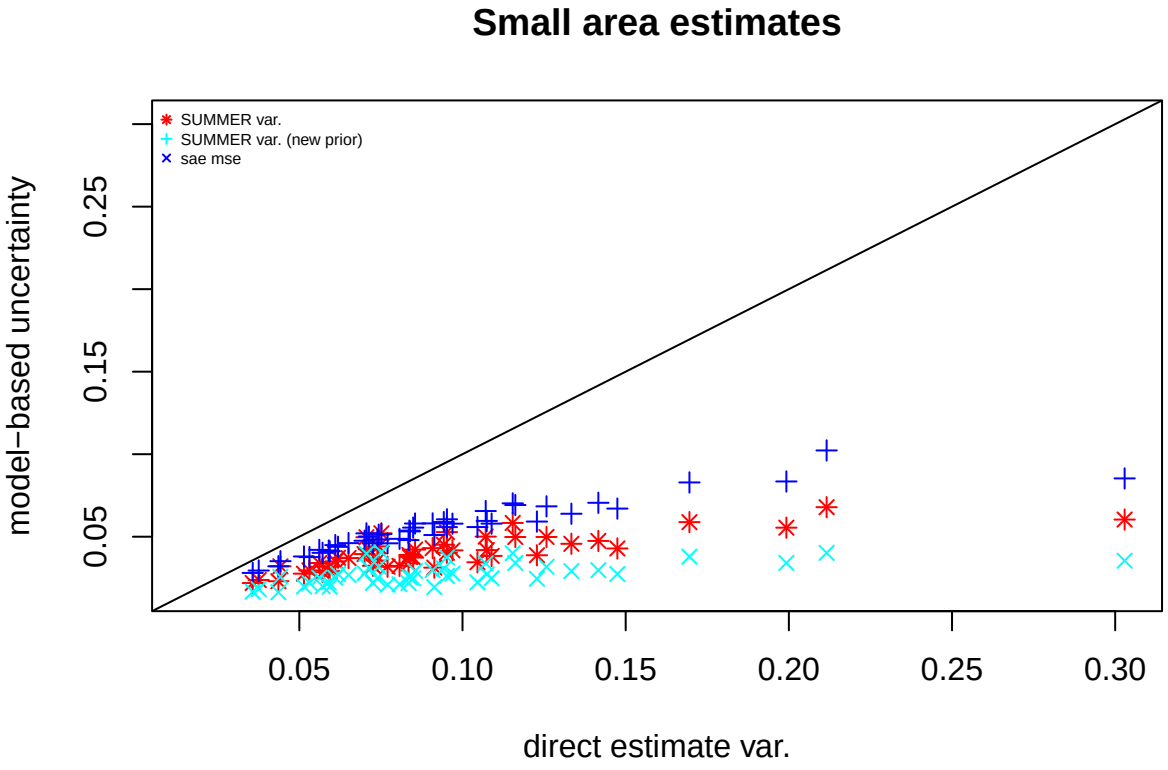
Comparison of location estimates

```
par(mfrow = c(1, 1))
range1 <- range(c(direct$diab2, toplot$mean.new))
plot(direct$diab2, toplot$mean, xlab = "direct estimates", ylab = "model-based estimates",
    main = "Small area estimates", col = "red", pch = 8, xlim = range1,
    ylim = range1)
points(direct$diab2, toplot$mean.sae, col = "blue", pch = 3)
points(direct$diab2, toplot$mean.new, col = "cyan", pch = 4)
legend("topleft", pch = c(8, 3, 4), col = c("red", "cyan", "blue"),
    legend = c("SUMMER", "SUMMER (new prior)", "sae"), bty = "n",
    cex = 0.5)
abline(0, 1)
```



Comparison of variability estimates

```
par(mfrow = c(1, 1))
range2 <- range(c(direct$logit.var, toplot$mse.sae, toplot$logit.var.new))
plot(direct$logit.var, toplot$logit.var, xlab = "direct estimate var.",
    ylab = "model-based uncertainty", main = "Small area estimates",
    col = "red", pch = 8, xlim = range2, ylim = range2)
points(direct$logit.var, toplot$mse.sae, col = "blue", pch = 3)
points(direct$logit.var, toplot$logit.var.new, col = "cyan",
    pch = 4)
legend("topleft", pch = c(8, 3, 4), col = c("red", "cyan", "blue"),
    legend = c("SUMMER var.", "SUMMER var. (new prior)", "sae mse"),
    bty = "n", cex = 0.5)
abline(0, 1)
```



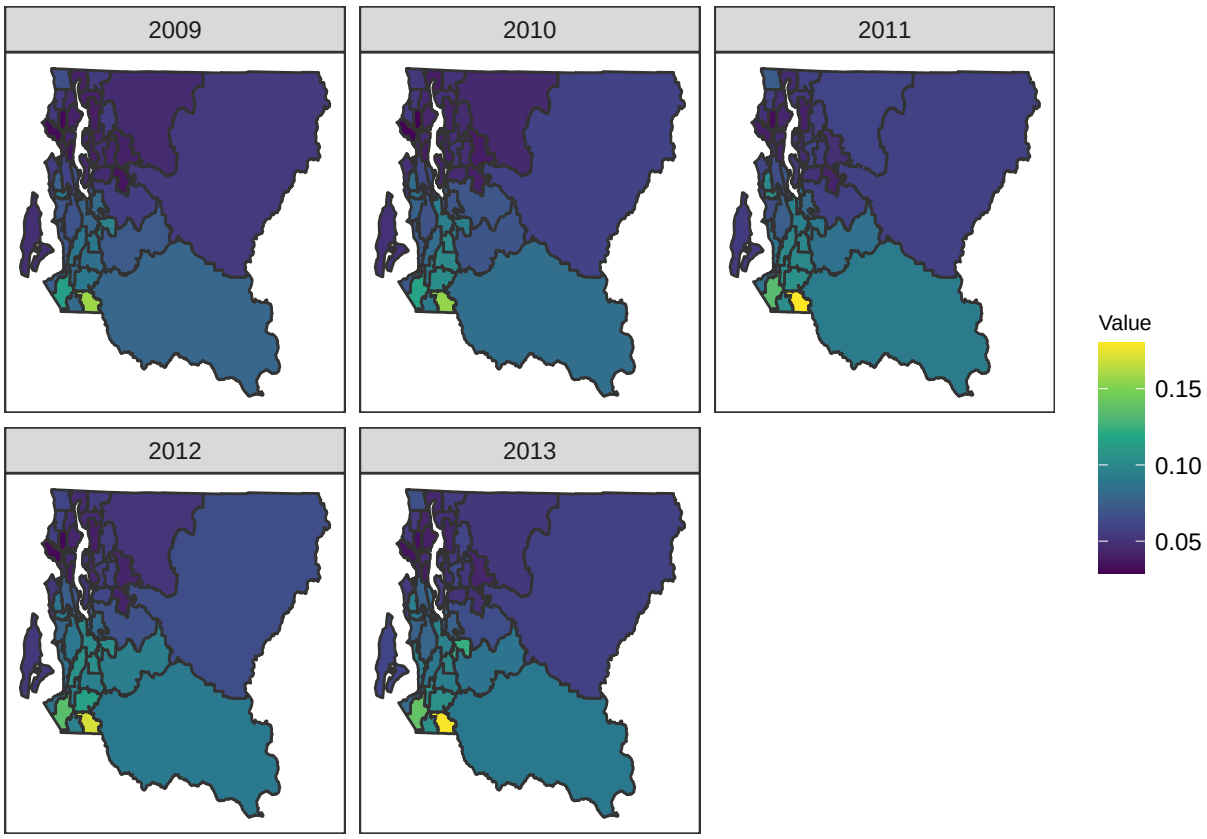
SAE in Space and Time

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 <- smoothSurvey(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:

https://arxiv.org/pdf/2007.05117.pdf