

# Lecture 4: The SUMMER Package

Jon Wakefield  
University of Washington

2019-05-22

# Small Area Estimation (SAE)

In this lecture, 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 BYM model
- Estimates from a smoothed Fay-Herriot model
- Estimates with different priors

# Load packages

We first load the SUMMER package.

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")
}
```

## Read in Data

BRFSS contains the full BRFSS dataset with 16,283 observations. The `-diab2` variable is the binary indicator of Type II diabetes - `strata` is the strata indicator and `- rwt_11cp` 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)
if (!isTRUE(requireNamespace("INLA", quietly = TRUE))) {
  install.packages("INLA", repos = "https://www.math.ntnu.no/inla/R/stable")
}
data(BRFSS)
BRFSS <- subset(BRFSS, !is.na(BRFSS$diab2))
BRFSS <- subset(BRFSS, !is.na(BRFSS$hracode))
```

## Create survey object

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.

```
data(KingCounty)
library(spdep)
nb.r <- poly2nb(KingCounty, queen = F, row.names = KingCounty$HRA2010v2_)
mat <- nb2mat(nb.r, style = "B", zero.policy = TRUE)
colnames(mat) <- rownames(mat)
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, ~hrcode, design, svymean)
head(direct, n = 2)
##                hrcode      diab2      se
## Auburn-North Auburn-North 0.1040315 0.02147752
## Auburn-South Auburn-South 0.2329329 0.04897800
```

# Binomial spatial smoothing model

We ignore the design and fit the model:

$$y_i | p_i \sim \text{Binomial}(n_i, p_i)$$
$$\theta_i = \log \left( \frac{p_i}{1 - p_i} \right) = \mu + \epsilon_i + S_i$$

with  $\epsilon_i \sim_{iid} N(0, \sigma_\epsilon^2)$  and  $[S_1, \dots, S_n]$  follow an intrinsic CAR (ICAR) model.

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

# The fitSpace function

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

```
smoothed <- fitSpace(data = BRFSS, geo = KingCounty,
  Amat = mat, family = "binomial", responseVar = "diab2",
  strataVar = NULL, weightVar = NULL, regionVar = "hracode",
  clusterVar = NULL, hyper = NULL, CI = 0.95)
head(smoothed$HT, n = 2)
##      HT.est    HT.sd HT.variance  HT.prec HT.est.original
## 1 -1.812902 0.1726995 0.02982513 33.52878    0.1402878
## 2 -1.196804 0.1760789 0.03100377 32.25414    0.2320442
##      HT.variance.original   n y      region
## 1           0.0004338385 278 39 Auburn-North
## 2           0.0009845287 181 42 Auburn-South
```



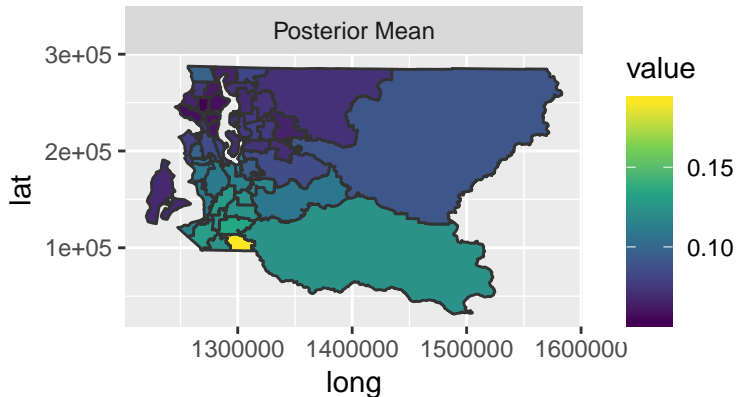
# The fitSpace function

The smoothed estimates of  $p_i$  and  $\theta_i$  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).

```
head(smoothed$smooth, n = 1)
##           region time      mean  variance  median    lower    upper
## 1 Auburn-North   NA -1.853919 0.01846505 -1.85323 -2.123352 -1.590408
##  mean.original variance.original median.original lower.original
## 1      0.1361808      0.0002546956      0.1355055      0.1068074
##  upper.original
## 1      0.1693841
head(smoothed$HT, n = 1)
##      HT.est    HT.sd HT.variance  HT.prec HT.est.original
## 1 -1.812902 0.1726995 0.02982513 33.52878      0.1402878
##  HT.variance.original  n y      region
## 1      0.0004338385 278 39 Auburn-North
```

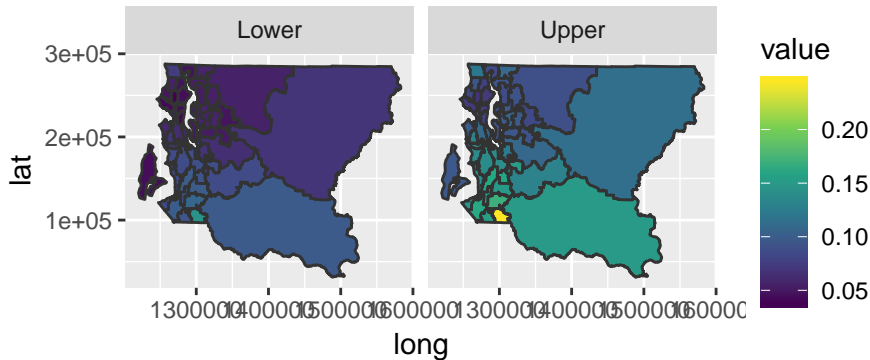
# Map the Posterior Mean Estimates

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



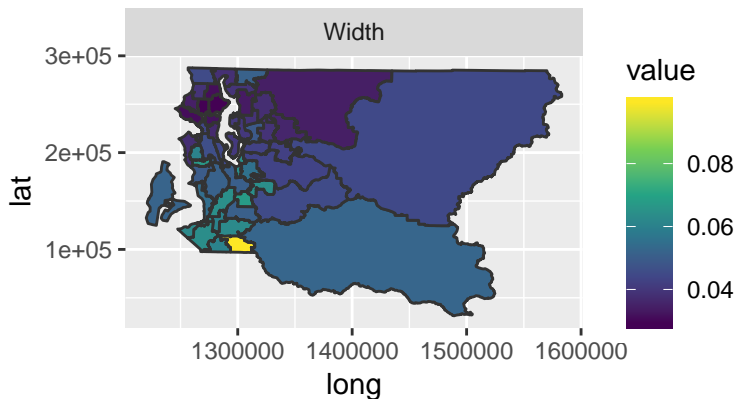
# Map the 2.5% and 97.5% Posterior Quantiles

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



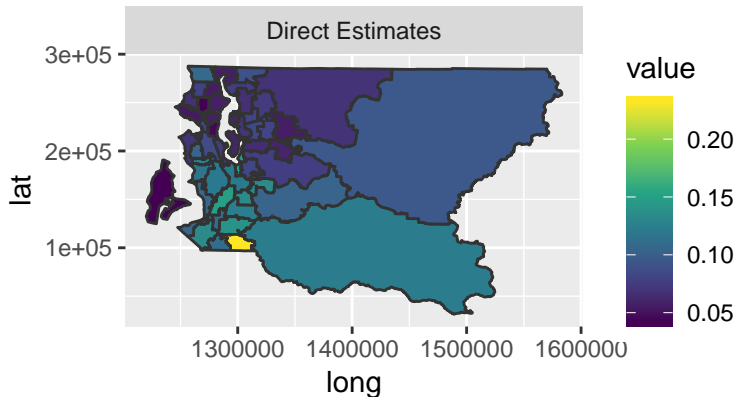
# Map the Interval Width

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



# Map the Direct Estimates

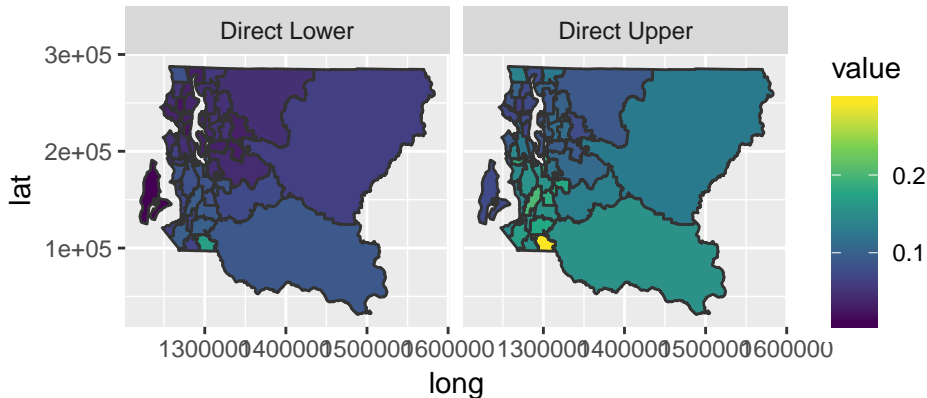
```
toplot$HTest <- smoothed$HT$HT.est.original
mapPlot(data = toplot, geo = KingCounty, variables = c("HTest"),
  labels = c("Direct Estimates"), by.data = "region",
  by.geo = "HRA2010v2_")
```



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

```
lo <- smoothed$HT$HT.est.original - 1.96 * sqrt(smoothed$HT$HT.variance.original)
hi <- smoothed$HT$HT.est.original + 1.96 * sqrt(smoothed$HT$HT.variance.original)
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 spatial smoothing model, but acknowledging the design

We now acknowledge the design and fit the model

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

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

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

with  $\epsilon_i \sim_{iid} N(0, \sigma_\epsilon^2)$  and  $[S_1, \dots, S_n]$  are ICAR.

```
svsmoothed <- fitSpace(data = BRFSS, geo = KingCounty,
  Amat = mat, family = "binomial", responseVar = "diab2",
  strataVar = "strata", weightVar = "rwt_llcp", regionVar = "hrcode",
  clusterVar = "~1", hyper = NULL, CI = 0.95)
```



## Comparison of Estimates: Setting Up

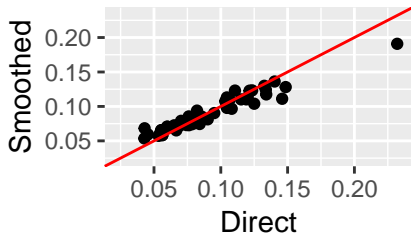
```
est <- data.frame(naive = smoothed$HT$HT.est.original,  
  weighted = svysmoothed$HT$HT.est.original, smooth = smoothed$smooth$mean.original,  
  weightedsmooth = svysmoothed$smooth$mean.original)  
var <- data.frame(naive = smoothed$HT$HT.variance.original,  
  weighted = svysmoothed$HT$HT.variance.original,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

# Comparison of Estimates: Setting Up

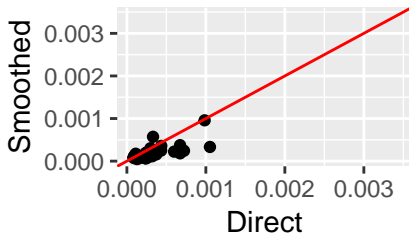
```
l1 <- range(est)
l2 <- range(var)
library(ggplot2)
g1 <- ggplot(est, aes(x = naive, y = smooth)) + geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  ggtitle("Naive Ests") + xlab("Direct") + ylab("Smoothed") +
  xlim(l1) + ylim(l1)
g2 <- ggplot(var, aes(x = naive, y = weightedsmooth))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

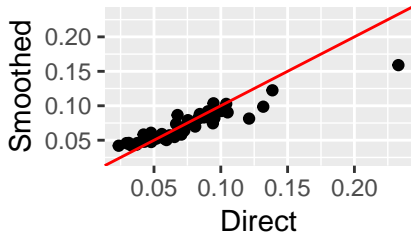
### Naive Ests



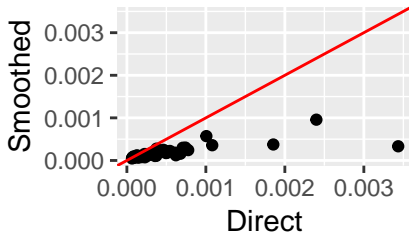
### Naive Vars



### Survey Wtd Ests



### Survey Wtd Vars



## Prior Choice

The `fitSpace` function has some default hyperprior choices built in.

- For Binomial models, we use  $\text{Ga}(0.5, 0.001488)$  on the latent precisions, which leads to a 95% prior interval for the residual odds ratio between  $[0.5, 2]$ .
- For Gaussian models we use the default  $\text{Ga}(1, 5 \times 10^{-5})$  prior from INLA.

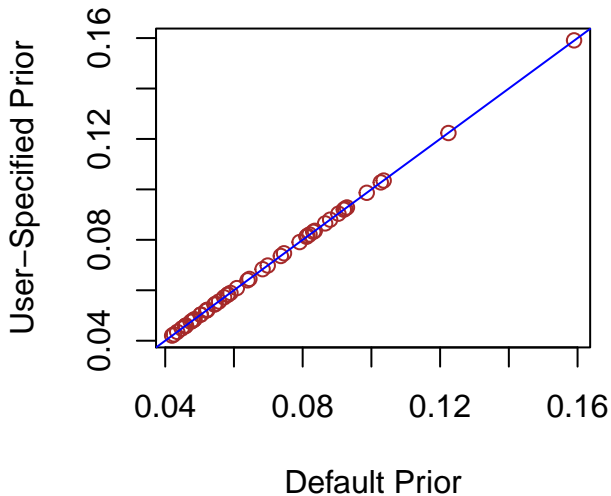
There are two ways to customize this default hyperprior choice. To simply update the hyper parameters of the Gamma prior, we can simply use the `hyper.besag` and `hyper.iid` arguments.

The other method is shown shortly.

# Fitting with a user-specified prior

```
svysmooth.1 <- fitSpace(data = BRFSS, geo = KingCounty,
  Amat = mat, family = "binomial", responseVar = "diab2",
  strataVar = "strata", weightVar = "rwt_llcp", regionVar = "hrcode",
  clusterVar = "~1", hyper = NULL, CI = 0.95, hyper.besag = c(0.5,
    0.01), hyper.iid = c(0.5, 0.01))
head(svysmooth.1$smooth, n = 1)
##           region time      mean  variance   median   lower   upper
## 1 Auburn-North  NA -2.178325 0.02915523 -2.179466 -2.510637 -1.839669
## mean.original variance.original median.original lower.original
## 1 0.1027678 0.0002480761 0.1015718 0.07516853
## upper.original
## 1 0.1369284
```

```
plot(svysmooth.1$smooth$mean.original ~ svysmoothed$smooth$mean.original,  
     xlab = "Default Prior", ylab = "User-Specified Prior",  
     col = "brown")  
abline(a = 0, b = 1, col = "blue")
```



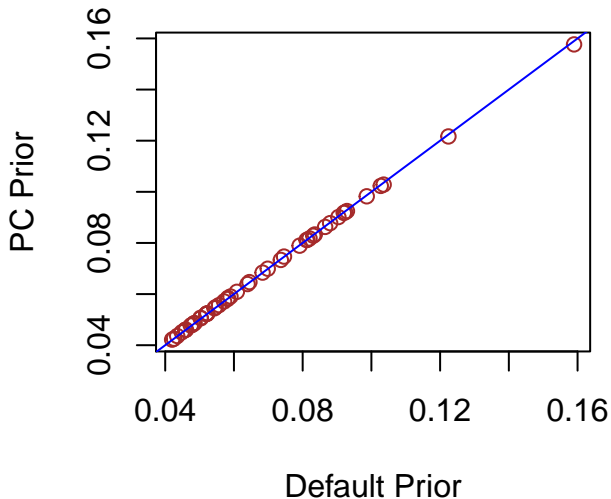
## Penalized Complexity (PC) Prior

To fit a PC prior we hard-code. For the BYM model the prior below states that:

- the prior probability that the total standard deviation exceeds 0.2/0.31 is 0.01, and that
- the prior probability that the proportion of the variance that is spatial is 0.5 is 2/3.

```
newformula <- "f(region.struct, model = 'bym2', graph = Amat,
  constr = TRUE, scale.model = TRUE, hyper = list(
  phi = list(prior = 'pc', param = c(0.5 , 2/3)
  , initial = -3),
  prec = list(prior = 'pc.prec', param = c(0.2/0.31 , 0.01)
  , initial = 5)))"
svsmooth.2 <- fitSpace(data = BRFSS, geo = KingCounty,
  Amat = mat, family = "binomial", responseVar = "diab2",
  strataVar = "strata", weightVar = "rwt_llcp", regionVar = "hrcode",
  clusterVar = "~1", hyper = NULL, CI = 0.95, newformula = newformula)
```

```
plot(svysmooth.2$smooth$mean.original ~ svysmoothed$smooth$mean.original,  
     xlab = "Default Prior", ylab = "PC Prior", col = "brown")  
abline(a = 0, b = 1, col = "blue")
```

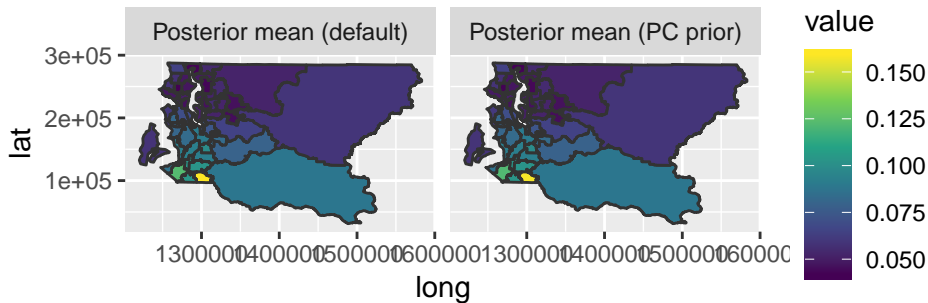




```

toplot <- svsmoothed$smooth
toplot$newprior <- svsmooth.2$smooth$mean.original
mapPlot(data = toplot, geo = KingCounty, variables = c("mean.original",
  "newprior"), labels = c("Posterior mean (default)",
  "Posterior mean (PC prior)", by.data = "region",
  by.geo = "HRA2010v2_")

```



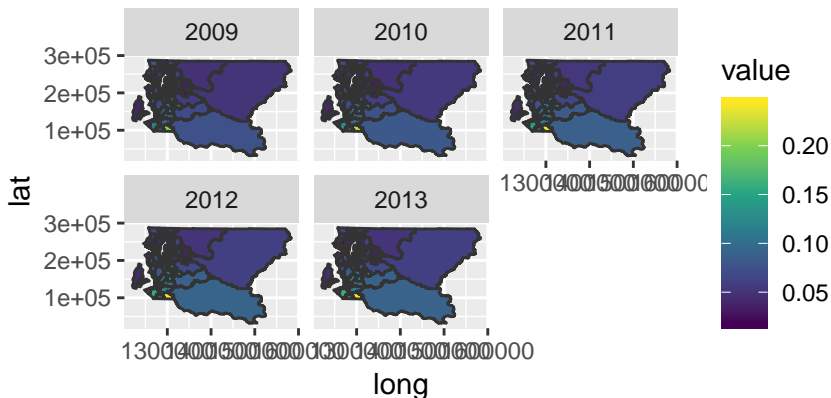
# SAE in Space and Time

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.

```
svysmoothed.year <- fitSpace(data = BRFSS, geo = KingCounty,  
  Amat = mat, family = "binomial", responseVar = "diab2",  
  strataVar = "strata", weightVar = "rwt_llcp", regionVar = "hrcode",  
  clusterVar = "~1", timeVar = "year", time.model = "rw1",  
  type.st = 4)
```

# Maps of Posterior Means over Time

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



# THE END!!

More materials can be found here:

<http://faculty.washington.edu/jonno/index.html>

If you want access to more materials, do get in touch:

[jonno@uw.edu](mailto:jonno@uw.edu)