# Small area estimation with R

eurostat

# Introduction

## What is a small area? A subpopulation/domain of interest for which

- few or even no observations are available
- a sampling design does not ensure sufficient data
- direct estimate, using only observations from the particular subpopulation, is insufficiently precise
- using a model that exploits similarities among many subpopulations can substantially improve on direct estimates

## Why study small area estimation?

- increasing demand for detailed, more specific figures, e.g. from local authorities
- we can often extract more information from data by combining data sources using model-based methods

# Design-based and model-based estimation

## Design-based / model-assisted estimation

- repeated sampling framework
- (inverse) inclusion probabilities calibrated s.t. $\sum_{i \in r} w_i x_i = t_x$
- multipurpose weights

$$\hat{t}_y = \sum_{i \in r} w_i y_i$$

## Model-based estimation

- predict for unobserved units based on model fit to data
- include design and non-response related information as covariates

$$\hat{t}_y = \sum_{i \in r} y_i + \sum_{i \in U \setminus r} \hat{y}_i$$

# Design-based domain estimation: direct estimates

'Direct': estimate depends only on observations $y_i$ in domain $d$:

$$\hat{\bar{Y}}_d = \sum_{i \in r_d} w_i y_i \Big/ \sum_{i \in r_d} w_i$$

**Pros:**

- single set of weights → simple and consistent
- mild dependence on (weighting) model

**Cons:**

- unsuitable for small $n_d = |r_d|$ due to large sampling variances
- in particular, no direct estimates possible for out-of-sample domains with $n_d = 0$

# SAE based on regression models

## A regression model without domain effects

- all differences between domains ‘explained’ by differences in (non-domain-specific) covariates
- ‘synthetic estimates’, underestimation of differences and uncertainty

## A regression model with domain effects

- domain-specific regression
- small sample sizes → noisy domain effect estimates
- similar to direct estimates

# Multilevel models

## Basic unit-level model

$$y_i \overset{\text{ind}}{\sim} \mathcal{N}(\beta' x_i + v_{d[i]}, \sigma^2) \quad \text{with} \quad v_d \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_v^2)$$

## Modeled / regularized domain-specific effects

- a.k.a. random (domain) effects, as opposed to 'fixed' regression effects
- for linear models best combination of synthetic and domain-specific regression estimates

$$\hat{\bar{Y}}_d \approx \gamma_d \underbrace{\left(\bar{y}_d + \hat{\beta}'(\bar{X}_d - \bar{x}_d)\right)}_{\text{survey regression est.}} + (1 - \gamma_d)\hat{\beta}'\bar{X}_d \quad \text{with} \quad \gamma_d = \frac{\sigma_v^2}{\sigma_v^2 + \sigma^2/n_d}$$

# R package hbsae

## Hierarchical Bayesian Small Area Estimation

- Basic area-level and unit-level models
- Model fitting: Bayesian using 1d numerical integration, or maximum likelihood
- Estimates and mean squared errors (MSEs) for domain means/totals
- Model comparison and evaluation: conditional AIC, cross-validation, residuals
- Benchmarking for consistency with published figures at aggregate level

# R package hbsae - usage

Unit-level model:

```
obj <- fSAE(y ~ x1 + x2, area="area", data=dat, Xpop=X)
```

Area-level model:

```
obj <- fSAE.Area(est.init, var.init, X)
```

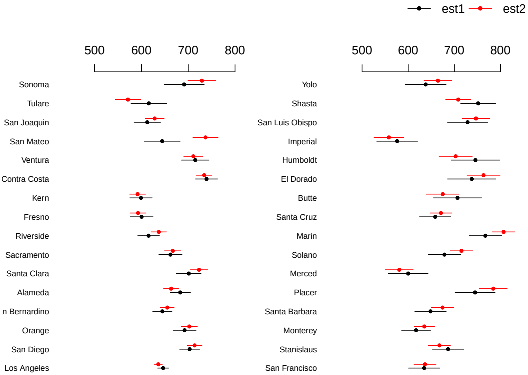
Result is an object of class sae with print, plot and other methods

# Example: basic unit-level model (hbsae)

```
library(hbsae)
library(survey)
data(api)
Xpop <- rowsum(model.matrix(~ stype + ell + meals, apipop), apipop$cname)
est <- fSAE(api00 ~ stype + ell, data=apisrs, area="cname", popdata=Xpop)
est
```

# Compare estimates (hbsae)

```
est1 <- fSAE(api00 ~ stype + ell, apisrs, "cname", Xpop)
est2 <- fSAE(api00 ~ stype + ell + mls, apisrs, "cname", Xpop)
plot(est1, est2)
```



# R package mcmcsae

## Markov Chain Monte Carlo Small Area Estimation

Simulation-based approach to Bayesian inference to support more complex multilevel models

- Sampling distributions for binary and count data, as well as for continuous data with outliers
- Multiple batches of modeled effects, e.g. nested random effects (say for province and municipality) crossed random effects (say for areas and time periods)
- Correlated random effects, exploiting ordering of domains, e.g. over time or geographically
- Non-normally distributed random effects, e.g. allowing for anomalous domains
- More options for model selection and diagnostics

# Workflow for estimation using mcmcsae

1 Specify model and set up a sampler function:

```
sampler <- create_sampler(formula, family="gaussian", formula.V, data, ...)
```

2 Fit the model using MCMC simulation to obtain draws from the parameters' posterior distribution:

```
sim <- MCMCsim(sampler)
```

3 Look at parameter posterior statistics and MCMC diagnostics:

```
summary(sim)
plot(sim, "sigma_")
```

4 Compute some model-diagnostics, e.g.

```
compute_DIC(sim)
```

5 Prediction/inference: generate simulation vectors for population quantities as an approximation to their posterior distribution

```
pred <- predict(sim, newdata=pop, fun.=sum)
summary(pred)
```

# Linear regression in mcmcsae: model component reg

Specify a linear regression model using the formula argument of create_sampler:

```
sampler <- create_sampler(y ~ x + f*z, data)
```

or, more generally, use

```
formula = y ~ reg(~ x + f*z, name="beta",
b0=0, Q0=0, ...)
```

- name for easy reference in the output (simulation) object
- b0 and Q0 for prior mean and precision Q0
- R,r and S,s for optional linear equality and inequality constraints $R\beta = r$ and $S\beta >= s$

Try ?reg to find out more

# Random effects in mcmcsae - model component gen

- formula: specify the effects (intercept, slopes) allowed to vary over the levels of factor
- factor: a formula defining the group levels over which the effects in formula are allowed to vary + specification of dependency structure among the levels, e.g. temporal or spatial
- var: variance structure among the effects in formula. Possible values: “scalar”, “diagonal”, “unstructured”.

Basic multilevel model: random intercepts for a single grouping variable

```
y ~ [fixed effects part] + gen(factor = ~ iid(domain))
```

# Usage example

log-transformed unit-level model, covariate x (and a constant), area random intercepts; prediction of area population means

```
sampler <- create_sampler(log(y) ~ x + gen(factor = ~ area), data=sam)
sim <- MCMCsim(sampler)
summary(sim)
pred <- predict(sim, newdata=pop,
                fun. = function(z) tapply(exp(z), pop$area, mean))
summary(pred)
```

# Some other R packages for SAE

## SAE specific

- emdi
- sae

## Multilevel

- lme4
- brms

## General

- rstan

# Further reading

- Rao and Molina, Small Area Estimation (2015, Wiley)
- McElreath, Statistical Rethinking: A Bayesian Course with Examples in R and STAN (2020, CRC)
- Gelman and Hill, Data Analysis Using Regression and Multilevel Hierarchical Models (2006, Cambridge)