Small area estimation II

Monica Pratesi and Caterina Giusti

Department of Economics and Management, University of Pisa

1st EMOS Spring School
Trier, Pisa, Manchester, Luxembourg, 23-27 March 2015
Structure of the Presentation

1 Introduction
2 Codes for GREG estimates
3 Codes for area level EBLUP estimates
4 Codes for M-quantile estimates
Part I

Small Area Estimation Software
Small Area Estimation Software

- One of the goal of the SAMPLE project (www.sample-project.eu) was to produce functions for small area estimation.
- The SAMPLE project partners involved in the small area estimation software development were:
  - Mixed model approach: University of Miguel Hernandez de Elche, University Carlos III de Madrid and marginally the University of Pisa
  - M-quantile approach: University of Pisa and University of Manchester/Southampton
- Basic small area estimation functions, such as the EBLUP, have been included in the SAMPLE Software Application.
- Functions are developed for the R statistical software.
Mixed model approach functions

Area-level small area estimation functions:
- Fay-Herriot model
- Area-level spatial model
- Area-level time model
- Area-level partitioned time models
- Area-level spatio-temporal models

Unit-level small area estimation functions:
- Unit-level time models
- EB prediction of poverty measures with unit level models
- Fast EB methods for estimation of fuzzy poverty measures
M-quantile approach functions

- M-quantile small area estimation of the mean
- Nonparametric M-quantile estimation of the mean
- M-quantile geographically weighted regression
- M-quantile CD estimators of the quantiles
- Nonparametric M-quantile CD estimators of the quantiles
- M-quantile poverty indicators estimators

Remark 1: The M-quantile approach can be used only as unit level model
Remark 2: The functions are written in R language so they are easy to modify but they are not fast!
Example of GREG estimates computed using R

- The R package JoSAE contains the function eblup.mse.f.wrap that can be used to obtain GREG (and also EBLUP) small area estimates.
- We use the data set on Norwegian forest data available in Breidenbach and Astrup (2012).
- The code below can be used to load the data and to plot the scatterplot of the $y$ and $x$ variables available in the sample data, as shown in the first set of slides.

```r
library(JoSAE)
data(JoSAE.sample.data)
plot(biomass.ha~mean.canopy.ht,JoSAE.sample.data)
```
Example of GREG estimates computed using R

- Then we can load also the population data and estimate a linear mixed model to obtain the predicted $y_{ij}$ values.

```r
data(JoSAE.domain.data)
fit.lme <- lme(biomass.ha ~ mean.canopy.ht, data=JoSAE.sample.data, random=~1|domain.ID))
```

where `biomass.ha` is the response variable, `mean.canopy.ht` is the auxiliary variable, `JoSAE.sample.data` is the data source and `random= 1|domain.ID` indicates that we are fitting a linear mixed model where the second level units are identified by `domain.ID`
Example of GREG estimates computed using R

- Then, it is necessary to check that the name of the auxiliary variable is the same in the population and sample datasets.
- If this is not the same, there is the need to change the name e.g. by converting the name from `mean.canopy.ht.bar` to `mean.canopy.ht`

```r
d.data <- JoSAE.domain.data
names(d.data)[3] <- "mean.canopy.ht"
```
Example of GREG estimates computed using R

Now we have all the information to obtain the GREG estimates by using the `eblup.mse.f.wrap` function

```r
results <- eblup.mse.f.wrap(domain.data = d.data, lme.obj = fit.lme)
```

The `eblup.mse.f.wrap` function has two arguments: `domain.data`, which contains the population data (in this case the dataset `d.data`) and `lme.obj`, which contains the fitted linear mixed model (in this case `fit.lme`)
Example of GREG estimates computed using R

- The `eblup.mse.f.wrap` function automatically produces several results, including GREG point and MSE small area estimates
- These results can be obtained by using the following commands:

```
results.GREG=cbind(results$GREG,results$GREG.se)
```

```
results.GREG

[,1]       [,2]
[1,] 112.97430     NA
[2,]  87.43037  22.3619200
[3,] 105.08065  24.9611244
[4,]  99.75545  0.6452861
[5,] 115.19719   8.6429206
[6,] 136.17706  16.9847138
[7,] 135.54343  14.8789830
[8,] 105.79197  15.4088365
[9,] 112.59132   7.1419053
[10,] 100.88560  12.3484579
[11,] 142.97128  24.7789172
[12,]  74.36564     NA
[13,] 124.35662     NA
[14,] 106.32493   8.3036947
```
Example of area-level EBLUP estimates computed using R

- The R library sae can be used to compute area-level and unit-level EBLUP small area estimates.
- We present here the code to obtain the estimates of mean grape production presented in the previous lesson.
- The code below can be used to load the data and to apply the eblupFH function.

```r
library(sae)
data(grapes)
```

- In the function we have to specify the formula of the mixed effect model (formula), the variance of the direct estimator (vardir), the estimation method, the maximum number of the iterations (MAXITER) and the tolerance (PRECISION).
Example of area-level EBLUP estimates computed using R

```r
resultREML <- eblupFH(formula=grapehect ~ area + workdays - 1, vardir=var,
data=grapes, MAXITER=500, PRECISION=1e-04)
```

The function returns a list with the following objects:

- **eblup:**
  - vector with the values of the estimators for the domains

- **fit**
  - **method:** type of fitting method applied ("REML", "ML" or "FH")
  - **convergence:** a logical value equal to TRUE if Fisher-scoring algorithm converges in less than MAXITER iterations
  - **iterations:** number of iterations performed by the Fisher-scoring algorithm
  - **estcoef:** a data frame with the estimated model coefficients in the first column (beta), their asymptotic standard errors in the second column (std.error), the t statistics in the third column (tvalue) and the p-values of the significance of each coefficient in last column (pvalue)
Example of area-level EBLUP estimates computed using R

The function returns a list with the following objects:

- **fit**
  - *refvar*: estimated random effects variance
  - *goodness*: vector containing three goodness-of-fit measures: loglikelihood, AIC and BIC

To print the results:

```r
resultMSE <- mseFH(grapehect ~ area + workdays - 1, var, data=grapes
resultMSE$mse
[1] 1.795904e+01 6.992134e+01 2.747896e+00 1.786207e+01 4.018609e+01 1.626821e-01 1.298132e+01 1.653500e+01 8.190804e+01 6.658895e-03 1.179526e+01...
```
Focus on R functions of M-quantile linear model estimators

Function to compute the small area averages, available on the SAMPLE project web-site:

mq.sae(y,x,x.outs,regioncode.s,regioncode.r,m,p,tol.value, maxit.value,k.value)

- **y**: the (numeric) response vector for sampled units
- **x**: a \( n \times p \) matrix of auxiliary variables which also has include a vector of ones for the intercept term
- **x.outs**: covariate information for out of sample units
- **regioncode.s**: area code for sampled units
- **regioncode.r**: area code for out of sample units
Focus on R functions of M-quantile linear model estimators

- $m$: the number of small areas
- $p$: size of $x + 1$ (including the intercept)
- $tol.value$: convergence tolerance limit for the M-quantile model. Default to 0.0001
- $maxit.value$: maximum number of iterations for the iterative weighted least squares. Default to 100
- $k.value$: tuning constant used with the Huber proposal 2 scale estimation. Default to 1.345
Focus on R functions of M-quantile linear model estimators

mq.sae function returns the following arguments:

- \textit{mq.cd}: estimates of small area means using the M-quantile Chambers-Dunstan estimator (Tzavidis et al. 2010)
- \textit{mq.naive}: estimates of small area means using the M-quantile naive estimator (Chambers and Tzavidis 2006)
- \textit{mse.cd}: MSE estimates for the M-quantile CD small area means
- \textit{mse.naive}: MSE estimates for the M-quantile naive small area means
- \textit{code.area}: the unique codes of the small areas
Example of the use of MQ.SAE.mean
Generating population data and drawing a sample (R commands)

```r
> # MQ-EBLUP
> source("c:\MQ_sae.R"); library(pps)
> sigmasq.u=3; sigmasq=6
> m=40
> ni=rep(5,m); Ni=rep(100,m); N=sum(Ni); n=sum(ni)
> set.seed(1973)
> u=rnorm(m,0,sqrt(sigmasq.u)); u=rep(u,each=100)
> e=rnorm(N, 0, sqrt(sigmasq))
> gr=rep(1:40,each=100)
> ar=unique(gr)
> uno=matrix(c(rlnorm(N,log(4.5)-0.5,0.5)),nrow=N,ncol=1)
> y=100+5*uno+u+e
> pop.matrix<-cbind(y,uno,gr); pop<-as.data.frame(pop.matrix)
> names(pop)<-c("y","x","area")
> # Drawing a sample
> s=stratsrs(pop$area,ni)
> x.lme=pop[s,]$x
> y.lme=pop[s,]$y
> regioncode.lme=pop[s,]$area
> pop.r<-pop[-s,]
```
Example of the use of MQ.SAE.mean
Example of R code for running function MQ.SAE.mean

tmp<-MQ.SAE.mean(y=y.lme,x=x.lme,regioncode.s=regioncode.lme,m=40, p=2,x.outs=pop.r[,2], regioncode.r=pop.r[,3],tol.value=0.0001, maxit.value=100,k.value=1.345)

Output of function MQ.SAE.mean

> tmp

mq.cd
[1] 115.7275 117.9384 115.3374 115.5339 116.3331 ...

mq.naive
[1] 115.9003 117.6583 115.0192 115.6947 116.0871 ...

mse.cd
[1] 0.55237498 0.80473242 1.54140859 0.75538562 2.13604316 ...

mse.naive
[1] 0.09710564 0.02790977 0.16425263 0.05226719 0.12559878...

code.area
[1] 1 2 3 4 5...
The MQPovertyLib package

- This package provides functions to estimate mean, quantiles and poverty indicators using the M-quantile approach
- It is think to complain with the EU-SILC survey data and with population data such as population censuses
- M-quantile model is estimated at household level but estimates are returned either at household level or at person level
- Poverty indicators computed are the Head Count Ratio (it measures the incidence of poverty) and the Poverty Gap (it measures the intensity of poverty)
- Authors: Stefano Marchetti, Nicola Salvati, Nikos Tzavidis and Caterina Giusti

Remark: the package is still under development and have no warranty and is released upon request to stefano.marchetti@unipi.it
The MQPovertyLib package

The package provides the following functions:

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQ-Poverty-Lib-package</td>
<td>Functions to estimate means, quantiles, HCRs and PGs for Small Areas using the MQ approach</td>
</tr>
<tr>
<td>income.example.sae</td>
<td>Simulated income data for 30 domains</td>
</tr>
<tr>
<td>mq.coef</td>
<td>It estimates the beta coefficient of each small area</td>
</tr>
<tr>
<td>MQ.SAE.mean</td>
<td>It estimates the small area mean</td>
</tr>
<tr>
<td>MQ.SAE.mean.pers</td>
<td>It estimates the small area mean at person (unit) level</td>
</tr>
<tr>
<td>MQ.SAE.poverty</td>
<td>It estimates the small area Head Count Ratio and the Poverty Gap indicators at household (cluster) level</td>
</tr>
<tr>
<td>MQ.SAE.poverty.persons</td>
<td>It estimates the small area Head Count Ratio and the Poverty Gap indicators at person level</td>
</tr>
<tr>
<td>MQ.SAE.poverty.smearing</td>
<td>It estimates the small area Head Count Ratio and the Poverty Gap indicators at household (cluster) level</td>
</tr>
<tr>
<td>MQ.SAE.quant</td>
<td>It Estimates the small area quantiles</td>
</tr>
<tr>
<td>MQ.SAE.quant.pers</td>
<td>It Estimates the small area quantiles at person level</td>
</tr>
<tr>
<td>QRLM</td>
<td>M-quantile linear regression model</td>
</tr>
</tbody>
</table>

There is an help for each function, accessible by the standard help R command (i.e. ?MQ.SAE.mean)