Vignette of the JoSAE package

Johannes Breidenbach*

6 October 2011: JoSAE 0.2

1 Introduction

The aim in the analysis of sample surveys is frequently to derive estimates of subpopulation characteristics. This task is denoted small area estimation (SAE) (Rao, 2003). Often, the sample available for the subpopulation is, however, too small to allow a reliable estimate. Frequently, auxiliary variables exist that are correlated with the variable of interest. Several estimators can make use of auxiliary information which may reduce the variance of the estimate (Rao, 2003). Another term for *small area* is *domain*. These two terms will be used interchangeable in the following.

The `JoSAE` package implements the *generalized regression* (GREG) (Särndal, 1984) and unit level *empirical best linear unbiased prediction* (EBLUP) (Battese et al., 1988) estimators and their variances. The *synthetic regression* and the *simple random sample* (SRS) estimates are also calculated. The purpose of the `JoSAE` package is to document the functions used in the publication of (Breidenbach and Astrup, 2011). The data used in that study are also provided.

If `R` is running, the `JoSAE` package can be installed by typing

```
> install.packages("JoSAE")
>
```

into the console[1].

The command

```
> library(JoSAE)
```

loads the package into the current workspace. We can get an overview of the packages' contents by typing

```
> ?JoSAE
```

2 Using the provided functions - small area estimates

For our small area estimates, we need

- sample data which contain the variable of interest and the auxiliary variables of all sampled population elements and
- domain data which contain the mean of the auxiliary variables of all population elements within each domain of interest. It is assumed that auxiliary information is available for every population element.

Both data sets need to have a corresponding domain ID.

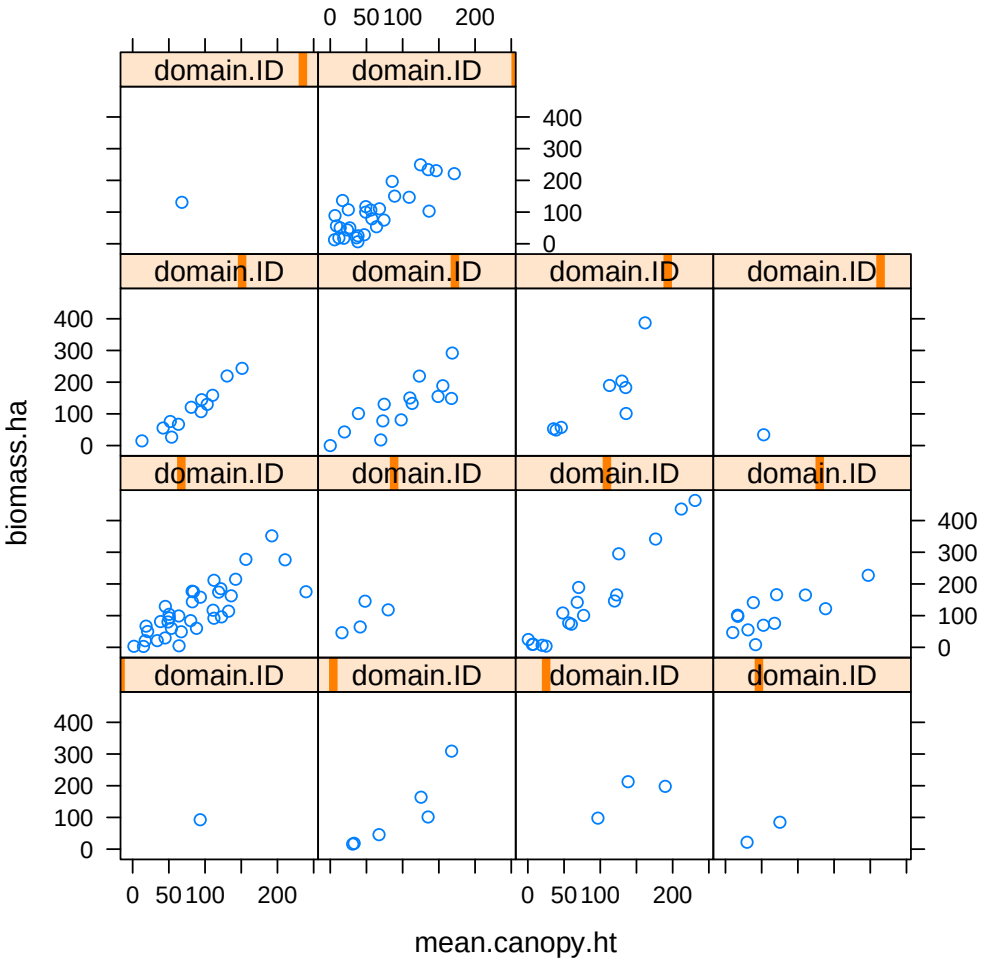
*Norwegian Forest and Landscape Institute, 1431 Ås, Norway, job@skogoglandskap.no, Tel.: +47 64 94 89 81

[1]The character ">" is not part of the command. A working Internet connection is required.

2.1 Mean forest biomass within Norwegian municipalities

To load and plot the data used by (Breidenbach and Astrup, 2011) we write:

```
>         #mean auxiliary variables for the populations in the domains
> data(JoSAE.domain.data)
>         #data for the sampled elements
> data(JoSAE.sample.data)
> #plot(biomass.ha~mean.canopy.ht,JoSAE.sample.data)
>
> library(lattice)
> print(xyplot(biomass.ha ~ mean.canopy.ht | domain.ID, data = JoSAE.sample.data))
```



The data set `JoSAE.sample.data` contains the above-ground forest biomass (the variable of interest) observed on sample plots of the Norwegian National Forest Inventory (NNFI) and the mean canopy height derived from overlapping digital aerial images (the auxiliary variable). The domain ID indicates in which of 14 municipalities (i.e., our small areas) the sample plot was located.

The data set `JoSAE.domain.data` contains the mean canopy height, photogrammetrically obtained from overlapping digital aerial images within the forest of a municipality. All population elements (i.e., not only those elements where field data from the NNFI were available) were used to derive this mean.

In order to make use of the auxiliary variables, a statistical model needs to be fit that links the variable of interest to the auxiliary variables. We fit a linear mixed-effects model (Pinheiro et al., 2011) with a random intercept on the municipality level to our data:

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

```
Linear mixed-effects model fit by REML
 Data: JoSAE.sample.data
       AIC      BIC    logLik
  1553.764 1565.616 -772.8822

Random effects:
 Formula: ~1 | domain.ID
        (Intercept) Residual
StdDev:    10.30361 49.85829

Fixed effects: biomass.ha ~ mean.canopy.ht
                  Value Std.Error  DF   t-value p-value
(Intercept)    6.694678  8.334032 130  0.803294  0.4233
mean.canopy.ht 1.375782  0.077531 130 17.744832  0.0000
 Correlation:
               (Intr)
mean.canopy.ht -0.754

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.12149463 -0.56323615 -0.05238025  0.55696863  3.11427777

Number of Observations: 145
Number of Groups: 14
```

In combination with the domain-level data, the functions provided in the `JoSAE` package can now be used to calculate domain level EBLUP estimates and their variances. Since the functions expect variable names in the domain data and the sample data to be the same, we first have to do some renaming:

```
>     #domain data need to have the same column names as sample data or vice versa
> d.data <- JoSAE.domain.data
> names(d.data)[3] <- "mean.canopy.ht"
```

Then we can use the `eblup.mse.f.wrap` function, which does all the work. This function is a wrapper function that calls several other `JoSAE` functions. All attributes the function needs are the domain data and the fitted model (an lme object).

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

Besides the EBLUP estimate and its variance, the function calculates the GREG and SRS estimate as well as a synthetic regression estimate based on a linear model fitted with the fixed-part of the lme formula. Many other domain characteristics are calculated by the `eblup.mse.f.wrap` function. The help page lists the details. Let's print some of the most interesting results in Tables 1 and 2.

domain.ID	N.i.domain	n.i.sample	sample.mean	GREG	EBLUP	Synth
1	105267	1	92.73	112.97	153.76	155.73
2	202513	6	109.06	87.43	107.82	113.81
3	134156	3	169.54	105.08	132.74	136.82
4	193807	2	53.29	99.76	123.88	126.45
5	1379945	35	118.39	115.20	118.49	124.05
6	176731	4	93.63	136.18	116.91	114.23
7	474615	17	152.52	135.54	117.73	105.72
8	442280	12	106.40	105.79	99.86	97.69
9	495568	12	113.70	112.59	116.84	119.66
10	520141	14	124.14	100.89	110.76	117.47
11	230756	8	152.95	142.97	135.89	133.98
12	83441	1	34.11	74.37	118.19	120.66
13	57858	1	130.78	124.36	95.01	94.67
14	905387	29	97.77	106.32	102.46	98.42

Table 1: Number of population and sampled elements as well as simple random sample, synthetic, GREG and EBLUP estimates of the mean above-ground forest biomass within 14 Norwegian municipalities.

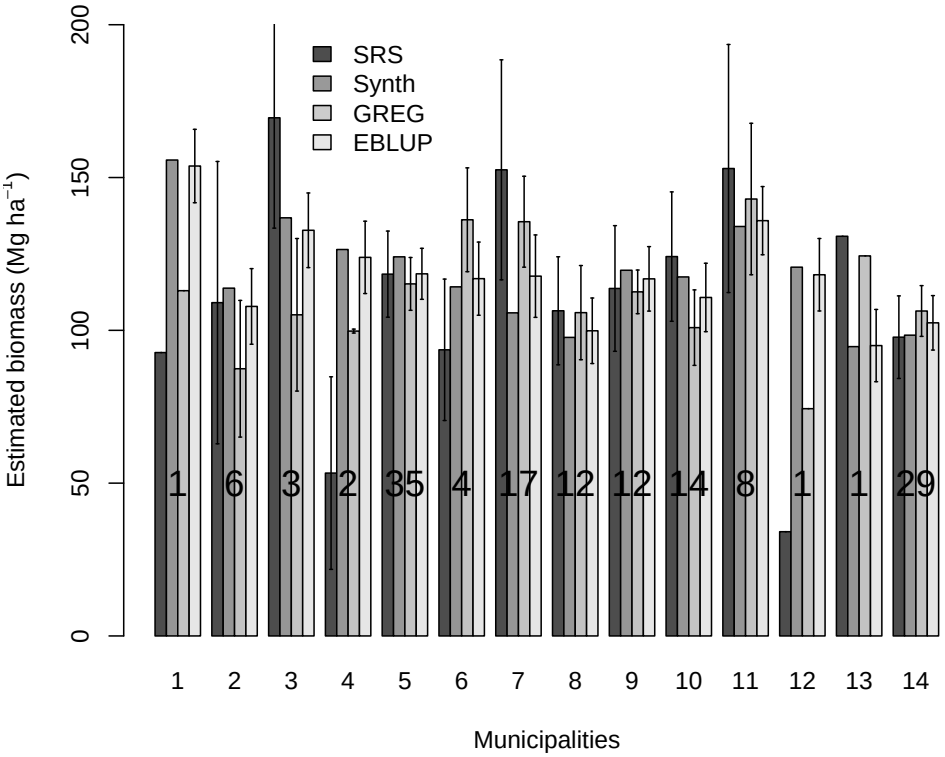
domain.ID	n.i.sample	sample.se	GREG.se	EBLUP.se.1	EBLUP.se.2
1	1			11.93	12.01
2	6	46.19	22.36	12.62	12.37
3	3	36.10	24.96	12.26	12.21
4	2	31.51	0.65	11.59	11.84
5	35	14.09	8.64	7.74	8.34
6	4	23.14	16.98	11.77	11.97
7	17	35.99	14.88	15.63	13.48
8	12	17.67	15.41	9.83	10.72
9	12	20.56	7.14	9.48	10.54
10	14	21.16	12.35	11.11	11.20
11	8	40.58	24.78	10.29	11.16
12	1			11.80	11.86
13	1			11.65	11.82
14	29	13.51	8.30	8.28	8.91

Table 2: Number of population and sampled elements as well as standard errors of the simple random sample, GREG and EBLUP estimates of the mean above-ground forest biomass within 14 Norwegian municipalities.

The `eblup.mse.f.wrap` function does not return a standard error for the synthetic regression estimate, since no estimators exist that consider its model bias. In Table 2, it needs to be noted that variances for the SRS and GREG estimates are unstable for small sample sizes within domains (say <6 observations). A variance estimate is technically impossible for domains with just one observation. The EBLUP variances are frequently smaller than the GREG variances and stable even for domains with just one observation. However, the EBLUP variance is model-based and thus relies on the correctness of the fitted model. Rao (2003) suggests two different EBLUP variances estimates. Both are returned by the `eblup.mse.f.wrap` function (Table 2).

The data can be visualized by:

```
> tmp <- result[,c("biomass.ha.sample.mean", "Synth", "GREG", "EBLUP")]
>     #actual plot
> tmp1 <- barplot(t(as.matrix(tmp)), beside=T
+             , names.arg=result$domain.ID
+             , xlab="Municipalities"
+                 , ylab=expression(paste("Estimated biomass (Mg ", ha^{-1}, ")" ))
+                 , ylim=c(0,200))
>     #print n.sample plots
> text(tmp1[2,]+.5, y = 50, labels = result$n.i.sample,cex=1.5)
>     #error bars
> tmp2<- result[,c("sample.se", "sample.se", "GREG.se", "EBLUP.se.2")]#sample.se twice to fill
> tmp2[is.na(tmp2)] <- 0
>         #plot error bars
>             #sample mean
> arrows(x0=tmp1[1,], y0=tmp[,1]+tmp2[,1], x1=tmp1[1,], y1 = tmp[,1]-tmp2[,1]
+             , length = 0.01, angle = 90, code = 3)
>             #GREG
> arrows(x0=tmp1[3,], y0=tmp[,3]+tmp2[,3], x1=tmp1[3,], y1 = tmp[,3]-tmp2[,3]
+             , length = 0.01, angle = 90, code = 3)
>             #EBLUP
> arrows(x0=tmp1[4,], y0=tmp[,4]+tmp2[,4], x1=tmp1[4,], y1 = tmp[,4]-tmp2[,4]
+             , length = 0.01, angle = 90, code = 3)
>     #legend
> legend(13,200, fill=grey(c(.3, .6, .8, .9)), legend=c("SRS", "Synth", "GREG", "EBLUP"), bty=
>
```



2.2 County crop areas in Iowa

Battese et al. (1988) were the first to describe the EBLUP estimator. They demonstrated its application using Landsat data to estimate the mean hectares of corn and soybeans within counties (small areas) in north-central Iowa. Thanks to Schoch (2011), the Landsat data are available in `R`. The functions in the `JoSAE` package should give approximately similar results as those presented by Battese et al. (1988) and Rao (2003, Table 7.3,p.144).

Let's get the data, split the data sets into a domain-specific and sample specific data frame and add a numeric domain ID to both. We will also exclude an "outlying" domain[2] in row 33 as was suggested by Battese et al. (1988):

```
> data(landsat)
>     #prepare the domain data - exclude "outlying" domain
> landsat.domains <- unique(landsat[-33,c(1, 7:8,10)])
>         #add a numeric domain ID
> landsat.domains$domain.ID <- 1:nrow(landsat.domains)
>         #change names to the names in the sample data
> names(landsat.domains)[2:3] <- c("PixelsCorn", "PixelsSoybeans")
>     #prepare the unit-level sample data
> tmp <- landsat[-33,c(2:6, 10)]
>         #add numeric domain ID
> landsat.sample <- merge(landsat.domains[4:5], tmp, by="CountyName")
>
```

Now we can fit a linear mixed-effects model and obtain our small area estimates:

```
> summary(landsat.lme <- lme(HACorn ~ PixelsCorn + PixelsSoybeans
```

[2]The `rsae` package was specifically developed for robust estimation where outliers do not need to be excluded. As of R 3.0.2, rsae is archived. Therefore, the landsat data were included in JoSAE.

```
+                             , data=landsat.sample
+                             , random=~1|domain.ID))

Linear mixed-effects model fit by REML
 Data: landsat.sample
       AIC      BIC    logLik
  308.3666 315.8492 -149.1833

Random effects:
 Formula: ~1 | domain.ID
        (Intercept) Residual
StdDev:    11.83317 12.13543

Fixed effects: HACorn ~ PixelsCorn + PixelsSoybeans
                  Value Std.Error DF   t-value p-value
(Intercept)    51.07040 24.409705 22  2.092217  0.0482
PixelsCorn      0.32872  0.049876 22  6.590780  0.0000
PixelsSoybeans -0.13457  0.055194 22 -2.438092  0.0233
 Correlation:
               (Intr) PxlsCr
PixelsCorn     -0.935
PixelsSoybeans -0.892  0.723

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-1.87576686 -0.70964548 -0.08543767  0.72472023  1.65660575

Number of Observations: 36
Number of Groups: 12

>     #obtain EBLUP estimates and MSE
> result <- eblup.mse.f.wrap(domain.data = landsat.domains
+              , lme.obj = landsat.lme)
>
```

County.name	n_i	EBLUP	EBLUP.se.1	EBLUP.se.2	GREG.se
Cerro Gordo	1	122.20	9.04	9.52	
Hamilton	1	126.22	9.04	9.46	
Worth	1	106.70	10.66	10.19	
Humboldt	2	108.44	8.11	8.18	19.88
Franklin	3	144.28	7.10	6.89	6.86
Pocahontas	3	112.14	6.68	6.70	6.65
Winnebago	3	112.80	6.62	6.66	9.13
Wright	3	122.00	6.29	6.55	8.73
Webster	4	115.33	5.95	5.92	4.09
Hancock	5	124.42	5.13	5.28	4.22
Kossuth	5	106.90	5.62	5.48	3.18
Hardin	5	143.01	5.57	5.63	5.06

Table 3: EBLUP estimates of county means of hectares under corn and estimated standard errors of the EBLUP and GREG estimates.

Comparing Table 3 with the reference (Battese et al., 1988; Rao, 2003) suggests that the results are quite similar but not exactly the same. The EBLUP estimates for the county means are slightly

different because Battese et al. (1988) adjusted the estimates to sum up to the approximately unbiased Survey-Regression estimate for the total area. The standard errors are slightly different since Battese et al. (1988) used the method of *fitting of constants* to estimate the model parameters but REML was used here. Finally, Battese et al. (1988) obtained standard errors also for Survey-Regression estimates within domains with just one observation. Unfortunately, no details were elaborated. Given that the Survey-Regression estimator should be the same as the GREG (Rao, 2003, p. 20), it is unclear to me how this was done (any hints would be appreciated).

All in all, it looks like the functions in the `JoSAE` package are correctly implemented.

3 Synthetic estimation

This section documents the estimators in Breidenbach, et al. (2015). R-code is given rather than implemented functions since the implementation is rather straight forward. The validation data are not given here except for one stand for which the variance estimation is explained.

It should again be noted that the synthetic estimators should be avoided, if observations are available within the small areas. This is because regression models can be biased for specific small areas.

Load NFI data, fit the linking model, and create data for one validation stand. Elev.Mean is the vegetation height, N and E are northing and easting.

```
> data(nfi)
>     #fit the model
> fit.nfi.iw <- lm(vol.2011~Elev.Mean, nfi, weights=1/Elev.Mean)
>   #data (model matrix, X) of one validation stand
> stand <- cbind(Intercept=1, Elev.Mean=c(147.41,127.48,98.66,118.85,124,120.81,119.7),
+                N=c(0,23,0,55,27,80,56), E=c(73,77,0,39,37,54,54) )
>   #aggregate to obtain X-bar
> stand.agg <- apply(stand[,1:2], 2, mean)
```

As indicated by one reviewer: If just variance estimator (3) is of interest, also a White estimator could be used. For all other estimators, a model for the residual variance is needed.

Synthetic variance estimators consider the uncertainty in the model. The uncertainty in the model parameters is the covariance matrix and will be called Sigma. The residual variance will be called sig. Due to the assumed heteroskedasticity, it needs to be multiplied with x_i to be meaningful.

```
>     #obtain covariance matrix
> Sigma <- vcov(fit.nfi.iw)
>     #residual variance
> sig <- summary(fit.nfi.iw)$sigma
```

Variance estimator (3) is based on the concept of the estimation of superpopulation parameters and can be obtained as follows for the example stand.

```
> var.p <- t(stand.agg) %*% Sigma %*% stand.agg
```

Variance estimator (5) does not make much sense for heteroskedastic models and is therefore not shown here. Variance estimator (6) can be implemented as:

```
> var.prh <- var.p + sum(sig^2 * stand[,2])/nrow(stand)^2
```

For variance estimator (7), we need some model that describes the spatial autocorrelation of grid cells within a stand. In the paper, we use one global model. This may not be the best solution, as it is likely that the structure of the autocorrelation is different from stand to stand. This is, however, out of the scope of the paper. A spatial range of 23 m was estimated for the spatial model based on the validation data. This process is not shown here. First we create the spatial object:

```
> library(nlme)
> spG <- corGaus(23, form = ~N+E)
```

Then we create the correlation matrix given the distance between the observations and the autocorrelation structure. Furthermore, we create the matrix of expected variances.

```
> cormat <- corMatrix(Initialize(spG, data.frame(stand)))
> varmat <- (sig * sqrt(stand[,2])) %o% (sig * sqrt(stand[,2]))
```

This finally results in estimator (7):

```
> var.prhs <- var.p + sum(cormat * varmat)/nrow(stand)^2
```

The square root of the variances (ignoring bias) results in the standard error (SE). The SE of the different estimators increases from (5)-(7) because more error components are considered.

```
> sqrt(c(var.p, var.prh, var.prhs))
```

```
[1] 10.69396 38.05104 47.23384
```

The larger the number of population elements (i.e., grid cells or pixels) is, the smaller will be the difference between the estimators. Again, beware of bias in synthetic estimates!

4 Acknowledgments

I would like to thank Tobias Schoch, the author of the `rsae` package (Schoch, 2011) for the provision of the Landsat data set.

References

Battese, G.E., R.M. Harter, and W.A. Fuller (1988): *An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data*, Journal of the American Statistical Association 83, pp. 28-36.

Breidenbach, J. and R. Astrup (2012): *Small area estimation of forest attributes in the Norwegian National Forest Inventory*. European Journal of Forest Research, 131:1255-1267.

Breidenbach, J. McRoberts, R. E., R. Astrup (2015): *Empirical coverage of model-based variance estimators for remote sensing assisted estimation of stand-level timber volume.* Remote Sensing of Environment. In press.

Pinheiro, J., D. Bates, S. DebRoy, D. Sarkar and the R Development Core Team (2011): *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-101.

Rao, J.N.K. (2003): *Small Area Estimation*, New Work: John Wiley and Sons.

Särndal, C. (1984): *Design-consistent versus model-dependent estimation for small domains*. Journal of the American Statistical Association, JSTOR, 624-631

Schoch, T. (2011): *rsae: Robust Small Area Estimation*, R package version 0.1-3.