

Small Area Estimation (Part II)

Short Course – Institute of Statistics, Republic of Albania



About

This slide deck has been prepared for the “Short Course on Small Area Estimation with R” at the Institute of Statistics of the Republic of Albania on February 17 and 18, 2022.

Contact

University of Applied Sciences Northwestern Switzerland
School of Business – Institute ICC
Prof. Dr. Tobias Schoch
Riggenbachstrasse 16
CH-4600 Olten
Switzerland

E-Mail: tobias.schoch@fhnw.ch

Phone: +41 62 957 21 02

Outline

1. Introduction
 2. Available R Packages
 3. SAIPE – Our Application – A Brief Brush-Up
 4. R Package sae
 5. R Package emdi
- Literature
- Appendix

1 Introduction

Overview of the slide deck

This part of the slide deck (i.e., Part II) is dedicated to the R packages and applications

- **R package overviews**

- CRAN Task View OfficialStatistics ([Link](#) or [Link](#))
- GitHub Awesome Official Statistics Software ([Link](#))

- **Relevant files for this part in the archive "SAEcourse.zip"**

- `snippets.R` (code snippets used in this slide deck)
- `methods.R` (utility functions that extend the R package `sae`; useful but not necessary)
- `application.md` and `application.pdf` (exercises and applications)

[In Part I of the slide deck (introduction), you can find instructions how to download the archive]

2 R Packages for SAE

sae – Small Area Estimation (Molina & Marhuenda, 2020)

- Models
 - Area-level model (Fay & Herriot and with spatial/ temporal correlations)
 - Unit-level model (Battese et al., 1988)
- Estimation methods (variance)
 - Maximum likelihood
 - Restricted maximum likelihood
 - Fay & Herriot
- MSE estimation
 - Analytic second-order approximation
 - Parametric bootstrap (and non-parametric bootstrap for spatial FH model)

The package is rather basic; there are only a few utility function

2 R Packages for SAE (ctd.)

emdi – Estimating and Mapping Disaggregated Indicators

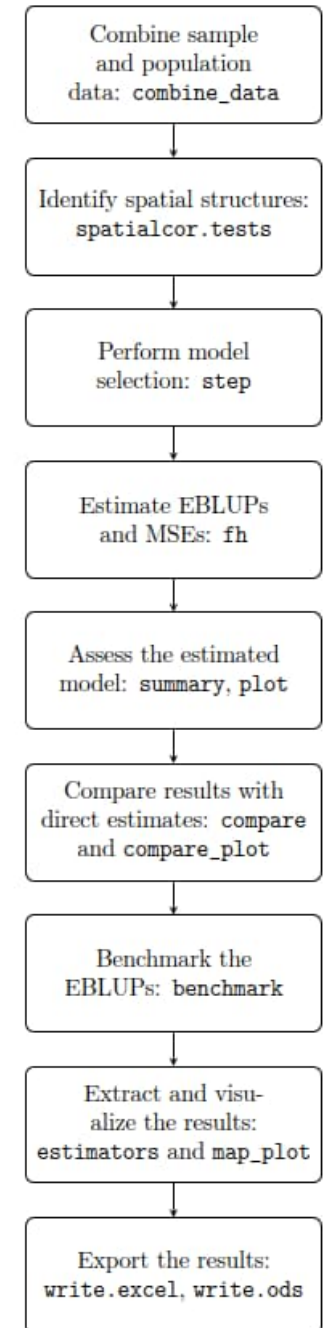
(Harmening et al., 2022; Kreutzmann et al., 2019)

- "A Framework for Producing Small Area Estimates Based on Area-Level Models in R"
 - Production framework
 - Wrapper (combines other packages)
 - Original package supported only the area-level model; since version 1.1.7 (March 2020) also unit-level model
- Focus on estimating indicators (not only mean and total)
- Methods
 - Basic models + spatial correlation + outlier robustness (FH model)
 - Transformations (log, arcsin, etc.) ⇒ important for handling indicators
 - Measurement errors (in the explanatory variables)

2 R Packages for SAE (ctd.)

emdi (ctd.)

- Package implements the entire "production chain": estimation, summary, diagnostic, prediction, benchmarks, maps, and export (to Microsoft Excel)
- Inspired by GSBPM: Generic Statistical Business Process Model (but does not follow GSBPM)
- Disadvantage
 - Large and rather convoluted package
 - It imports dozens of packages (and their dependencies)
 - For the accompanying article (J Stat Software), the authors relied on 136 packages (in my humble opinion: this is way too much)



2 R Packages for SAE

rsae – Robust Small Area Estimation (Schoch, 2014)

- Models
 - Unit-level model (Battese et al., 1988)
- Estimation methods (variance and regression)
 - Maximum likelihood
 - Huber M-estimator
- MSE estimation
 - Parametric bootstrap
- Unkept promise
 - "Robust methods for area-level model will be implemented" (claim) ...
 - This was not case (yet!). The code base is ready but not tested.

3 SAIPE – Our Application – Brief Brush-Up

- **Goal:** Estimates of child poverty rates for the 51 US states (2005)
- CPS data (survey) \Rightarrow direct estimate of poverty rate (rather unreliable)
- Administrative data (tax, census) \Rightarrow auxiliary information (model)

> `source("saipe.R")` # data are stored as code

> `head(dat, 3)`

	prIRS	nfIRS	prCensus	yi	vi	state
1	23.1582	14.6657	11.8464	19.4400	3.0371	AL
2	15.1852	10.9282	7.6478	11.0042	2.2330	AK
3	19.5926	19.3337	9.0293	17.4417	2.6003	AZ

Explanatory variables

prIRS poverty rate (tax data)
 nfIRS non filer rate (tax data)
 prCensus poverty rate (census)

Survey estimates (CPS data)

yi direct estimator of poverty rate
 vi variance of direct estimator

3 SAIPE – Our Application (ctd.)

- **Model**

Fay-Herriot model (for $i = 1, \dots, 51$ states)

$$y_i = \beta_0 + \beta_1 \text{prIRS}_i + \beta_2 \text{nfIRS}_i + \beta_3 \text{prCensus}_i + u_i + e_i,$$

where

- $u_i \sim N(0, v_i)$, the v_i 's (variance of direct estimator) are known
- $e_i \sim N(0, \sigma^2)$, the variance σ^2 is unknown

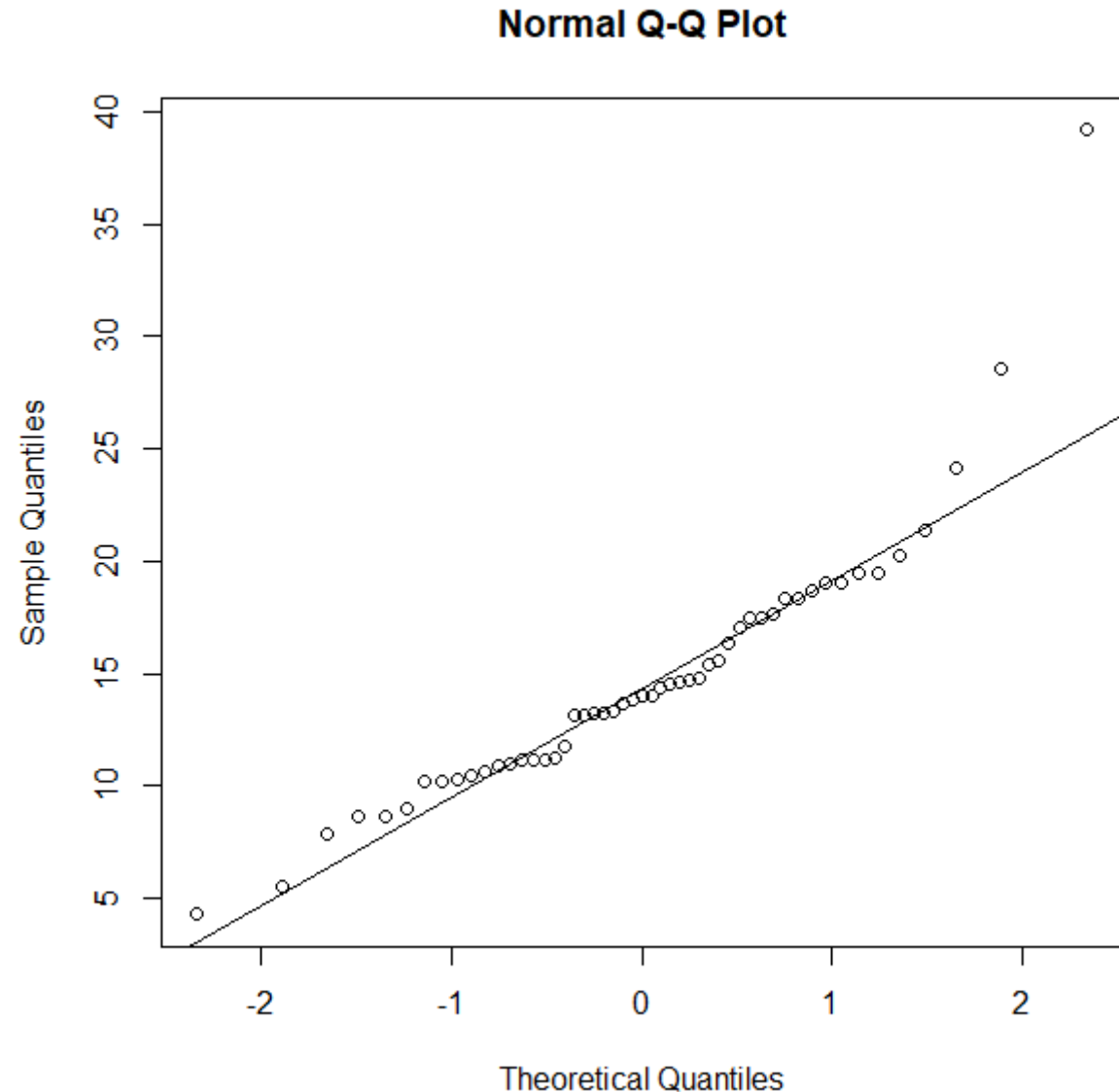
- **Estimators**

- $\beta = (\beta_0, \dots, \beta_3)^T$ is estimated by weighted least squares
- σ^2 is estimated by REML

3 SAIPE – Our Application (ctd.)

Normality assumption

- $u_i \sim N(0, v_i)$ cannot be checked (it is plausible by the central limit theorem)
- $e_i \sim N(0, \sigma^2)$ can be checked having fitted the model (\Rightarrow diagnostics, later)
- At this stage, we can study the distribution of the direct estimators y_i
`qqnorm(dat$yi)` and `qqline(dat$yi)`



3 SAIPE – Our Application (ctd.)

Some general remarks

- We are grateful to **William R. Bell (US Census Bureau)** for making the 2005 SAIPE data available.
- For more current datasets see <https://www.census.gov/programs-surveys/saipe.html>
- The SAIPE data are **"ready to use"** because the US Census Bureau prepared and compiled the data.
 - Computation of the direct estimator and variances (CPS survey)
 - The variances have been processed (generalized variance functions)
 - The US Census Bureau selected the auxiliary datasets (Census, IRS tax data) and the relevant variables
- **In practice, we need to select the potential datasets and variables** (and study them in the context of our model)

4 R Package sae

Step 0. Load the package

```
> library(sae)
```

Step 1. Fit the Fay-Herriot model

```
> m <- eblupFH(yi ~ prIRS + nfIRS + prCensus, vardir = vi,  
              data = dat)
```

- By default, method = "REML" (alternatives: "ML" or "FH")
- Argument var specifies the variance of the direct estimator (the v_i 's)
- The function returns a list with slots
 - eblup (matrix with estimates $\hat{\theta}_i^{\text{EBLUP}}$ for $i = 1, \dots, n$)
 - fit (list with entries method, convergence, iterations, estcoef, refvar, goodness)

4 R Package sae (ctd.)

Step 1. Fit the Fay-Herriot model (ctd.)

```
> m$fit
```

Part 1	Part 2
\$method	\$estcoef (estimated coefficients $\hat{\beta}$)
[1] "REML"	beta std.error tvalue pvalue
	(Intercept) -4.156450 1.5339727 -2.70999 6.7364e-03
\$convergence	prIRS 0.226095 0.1512419 1.49426 1.3493e-01
[1] TRUE	nfIRS 0.870038 0.1435364 6.06150 1.3489e-09
	prCensus 0.436531 0.1817465 2.40170 1.6311e-02
\$iterations	\$refvar
[1] 6	[1] 3.922967 (estimated variance $\hat{\sigma}^2$)
	\$goodness
	loglike AIC BIC KIC
	-118.1490 246.2980 255.9571 251.2980

4 R Package sae (ctd.)

Step 1. Fit the Fay-Herriot model (ctd.)

```
> mysummary(m)
```

	Estimate	Std.Err	t value	Pr(> t)	
(Intercept)	-4.15645	1.53397	-2.7096	0.006736	**
prIRS	0.22610	0.15124	1.4949	0.134934	
nfIRS	0.87004	0.14354	6.0615	1.349e-09	***
prCensus	0.43653	0.18175	2.4019	0.016312	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

The `mysummary()` function is not part of the `sae` package; see `methods.R`

4 R Package sae (ctd.)

Step 2. Model selection: Information criteria

```
> m$fit$goodness # main model
```

loglike	AIC	BIC	KIC
-118.1490	246.2980	255.9571	251.2980

```
> eblupFH(yi ~ nfIRS + prCensus, vardir = vi, # 2nd model
           data = dat)$fit$goodness
```

loglike	AIC	BIC	KIC
-119.3031	246.6062	254.3335	250.6062

- In terms of AIC, the main model is superior. Regarding BIC and KIC, the 2nd model is better \Rightarrow not conclusive
- See Marhuenda et al. (2014) for more on information criteria

4 R Package sae (ctd.)

Step 3. Rudimentary diagnostics

```
> tmp <- myFH(yi ~ prIRS + nfIRS + prCensus, vardir = vi,  
              data = dat)
```

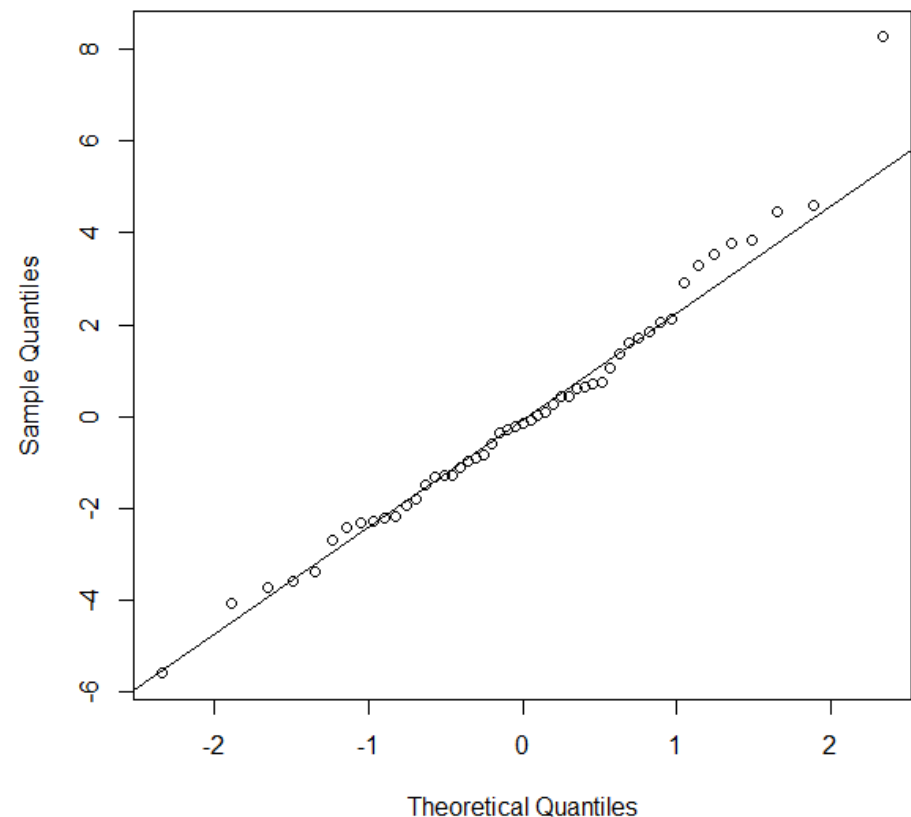
```
> qqnorm(tmp$fit$residuals)
```

```
> qqline(tmp$fit$residuals)
```

- QQ-plot of the standardized residuals
- **NOTE!** We use function `myFH()` in place of `eb1upFH()`; thus, we can extract the residuals; the function `myFH()` is not part of the `sae` Package; see `methods.R`



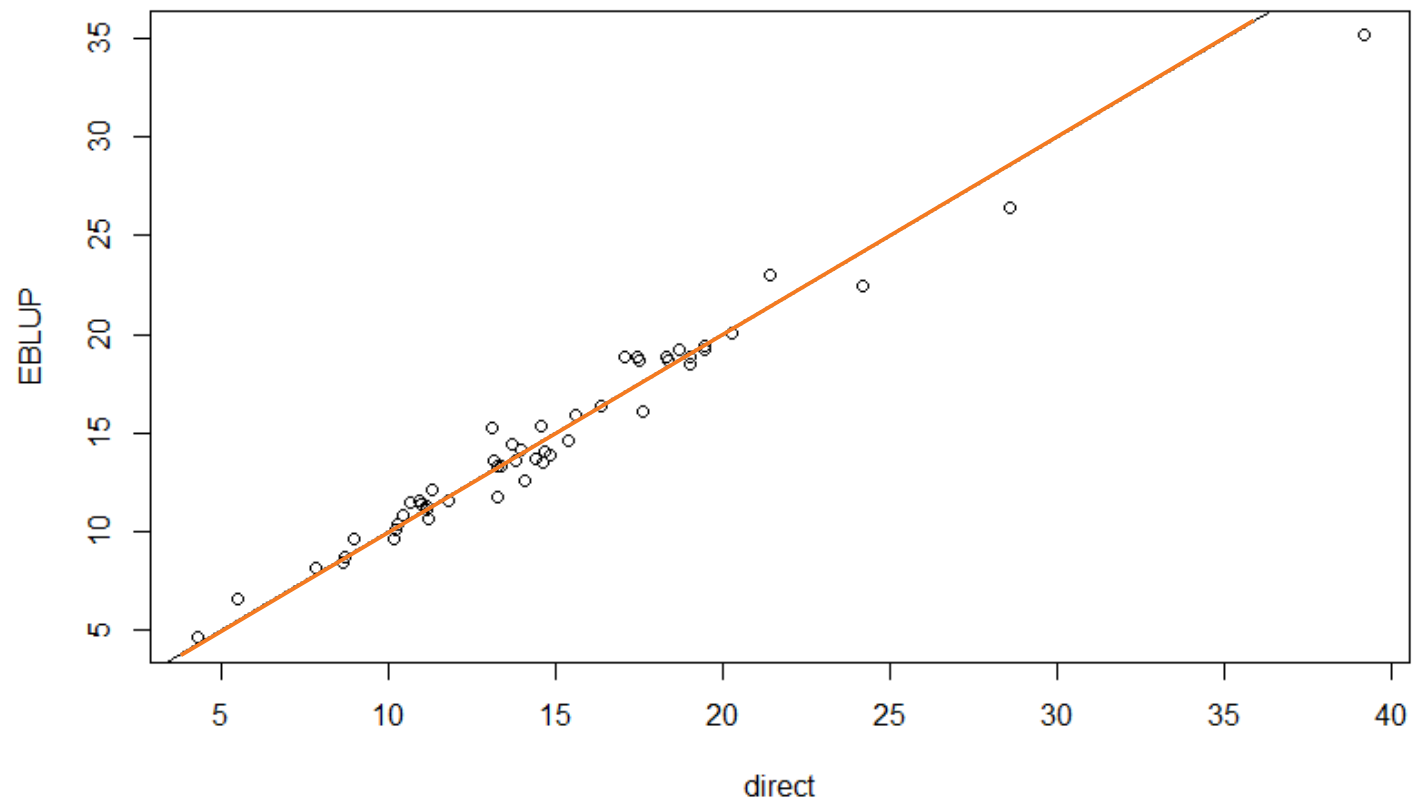
Normal Q-Q Plot



4 R Package sae (ctd.)

Step 3. Rudimentary diagnostics (ctd.)

```
> plot(dat$yi, m$eblup[,1], xlab = "direct", ylab = "EBLUP")  
> abline(0, 1) # 45-degree line
```



This plot is also
available for the
eblupFH() function

4 R Package sae (ctd.)

Step 4. MSE estimation

```
> mse_analytic <- mseFH(yi ~ prIRS + nfIRS + prCensus,  
                        vardir = vi, data = dat)
```

- By default, $B = 0$ (number of bootstrap replicates) \Rightarrow analytical MSE
- If $B > 0$, then B bootstrap replicates are used to estimate the MSE

```
> mse_bootstrap <- mseFH(yi ~ prIRS + nfIRS + prCensus,  
                        vardir = vi, B = 500, data = dat)
```

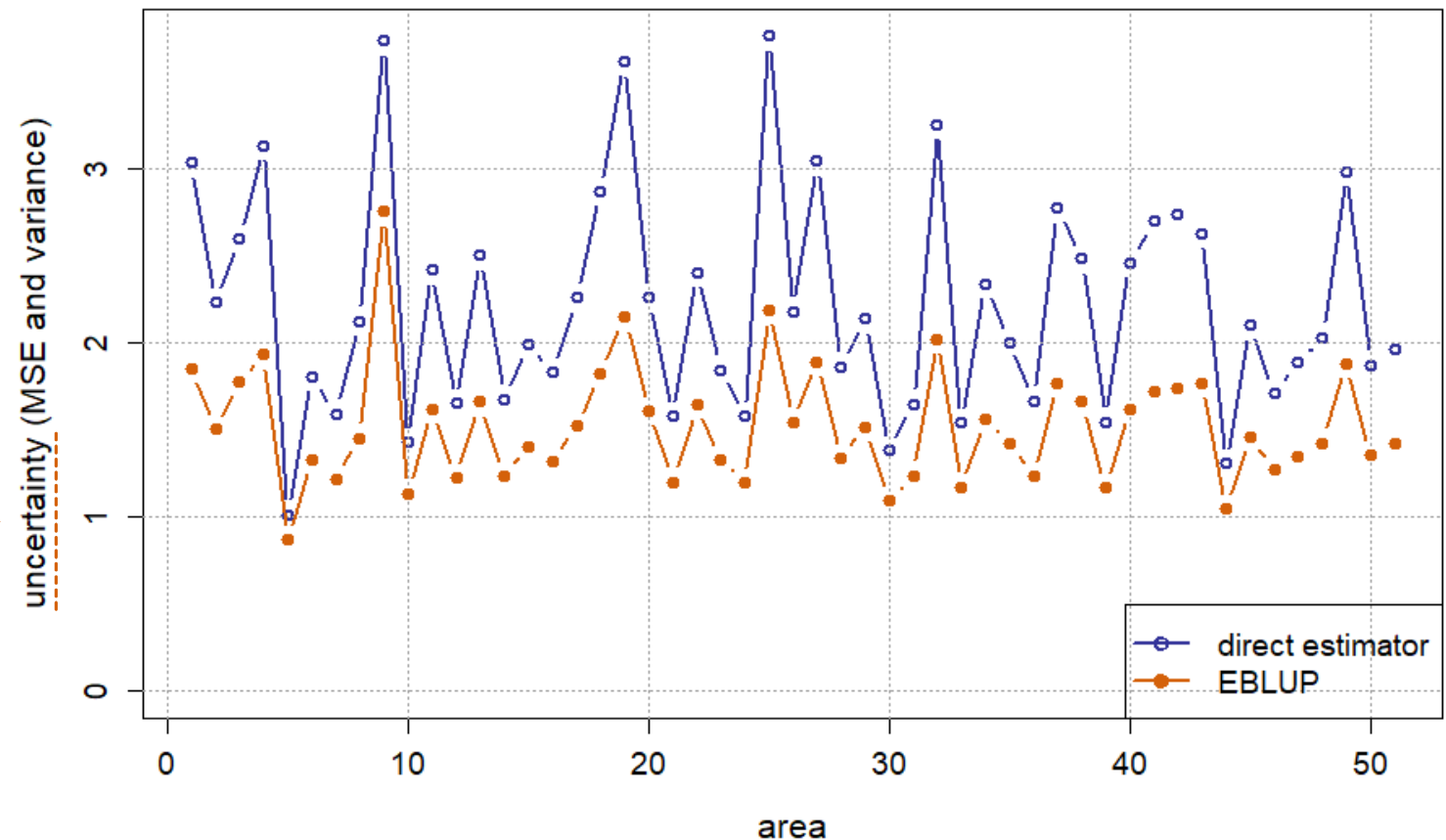
- $B = 500$ bootstrap replicates is about the minimum we should use

4 R Package sae (ctd.)

Step 4. MSE estimation (ctd.)

```
> plot(dat$vi)
> lines(mse_analytic$mse)
```

Comparing
MSE with
variance!



4 R Package sae (ctd.)

Step 5. Confidence intervals

Normal-theory 95% confidence intervals of the EBLUP (using estimated MSE)

```
> est <- data.frame(yi = dat$yi, EBLUP = m$eblup[, 1])
> alpha <- 0.05 # 5% level of significance
> est$ci_low <- est$EBLUP - sqrt(mse_analytic$mse) *
  qnorm(1 - alpha/2)
> est$ci_high <- est$EBLUP + sqrt(mse_analytic$mse) *
  qnorm(1 - alpha/2)
> head(est, 3)
```

	yi	EBLUP	ci_low	ci_high
1	19.4400	19.25261	16.583505	21.92172
2	11.0042	11.41015	9.005141	13.81515
3	17.4417	18.87445	16.261264	21.48764

5 R Package emdi

Package installation (Vers. 2.11): 88 dependencies...

```
package 'bit' successfully unpacked and MD5 sums checked
package 'prettyunits' successfully unpacked and MD5 sums checked
package 'rprojroot' successfully unpacked and MD5 sums checked
package 'rstudioapi' successfully unpacked and MD5 sums checked
package 'colorspace' successfully unpacked and MD5 sums checked
package 'utf8' successfully unpacked and MD5 sums checked
package 'bit64' successfully unpacked and MD5 sums checked
package 'progress' successfully unpacked and MD5 sums checked
package 'brew' successfully unpacked and MD5 sums checked
package 'commonmark' successfully unpacked and MD5 sums checked
...
package 'emdi' successfully unpacked and MD5 sums checked
```

5 R Package emdi

Step 0. Load the package

```
> library(emdi)
```

Step 1. Fit the Fay-Herriot model

```
> m <- fh(yi ~ prIRS + nfIRS + prCensus, vardir = "vi",  
          combined_data = dat, method = "reml")
```

Empirical Best Linear Unbiased Prediction (Fay-Herriot)

Out-of-sample domains: 0

In-sample domains: 51

Variance and MSE estimation:

Variance estimation method: reml

Variance of random effects: 3.922974

MSE method: no mse estimated

Transformation: No transformation

5 R Package emdi

Step 1. Fit the Fay-Herriot model (ctd.)

```
> head(estimators(m), 3)
```

	Domain	Direct	FH	Out
1	1	19.4400	19.25261	0
2	2	11.0042	11.41015	0
3	3	17.4417	18.87445	0

```
> summary(m)
```

Call:

```
fh(fixed = yi ~ prIRS + nfIRS + prCensus, vardir = "vi",  
   combined_data = dat, method = "reml")
```


5 R Package emdi

output of summary (ctd.)

Out-of-sample domains: 0

In-sample domains: 51

Variance and MSE estimation:

Variance estimation method: reml

Estimated variance component(s): 3.922974

MSE method: no mse estimated

Coefficients:

	coefficients	std.error	t.value	p.value	
(Intercept)	-4.15645	1.53397	-2.7096	0.006736	**
prIRS	0.22610	0.15124	1.4949	0.134934	
nfIRS	0.87004	0.14354	6.0614	1.349e-09	***
prCensus	0.43653	0.18175	2.4019	0.016312	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

5 R Package emdi

output of summary (ctd.)

Explanatory measures:

	loglike	AIC	BIC	KIC	R2	AdjR2
1	-118.149	246.298	255.9571	251.298	0.7880858	0.8419033

Residual diagnostics:

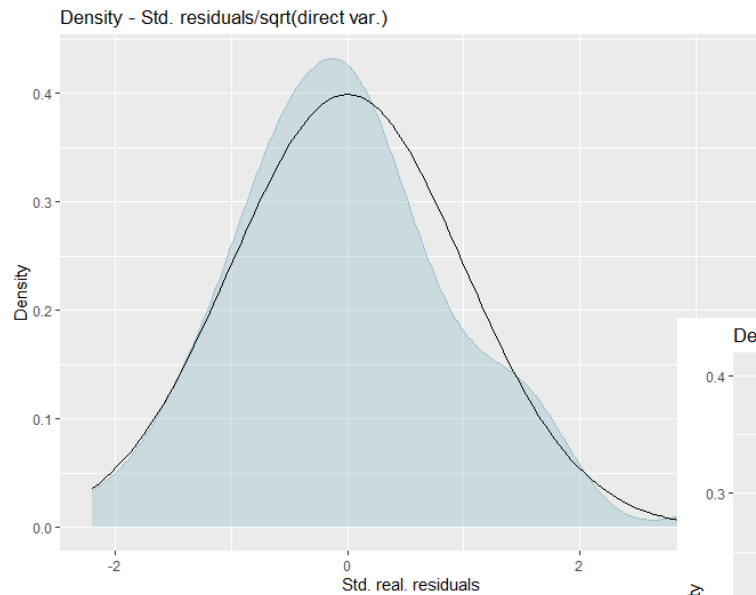
	Skewness	Kurtosis	Shapiro_W	Shapiro_p
Standardized_Residuals	0.6342088	4.100074	0.9718588	0.2637918
Random_effects	0.3664708	3.035362	0.9869228	0.8424495

Transformation: No transformation

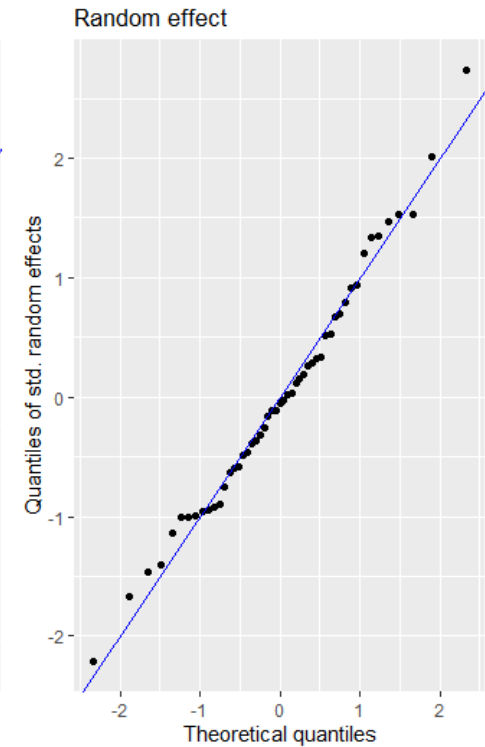
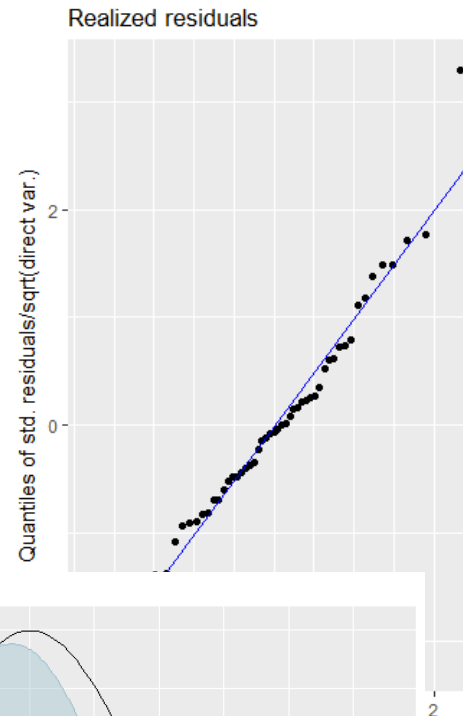
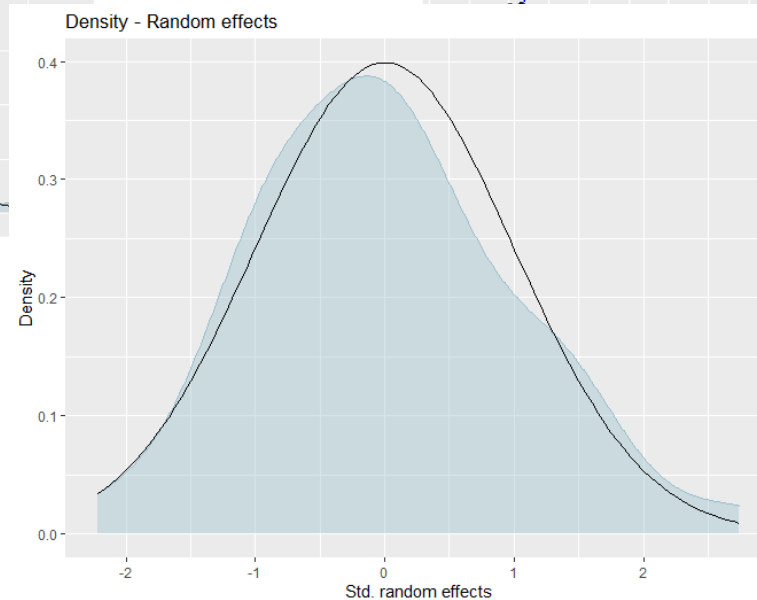
5 R Package emdi

Step 3. Diagnostics

```
> plot(m)
```



Density
plots



Residuals

5 R Package emdi

Step 4. MSE estimation

```
> m <- fh(fixed = yi ~ prIRS + nfIRS + prCensus, vardir = "vi",  
         combined_data = dat, method = "reml"  
         MSE = TRUE, mse_type = "analytical")
```

Alternative MSE estimators

- `mse_type = "jackknife"`
- `mse_type = "weighted_jackknife"`
- `mse_type = "boot"`
- ...

Plot method (shows direct vs EBLUP and MSE by area)

```
> compare_plot(m)      # not shown
```

5 R Package emdi

Step 5. Miscellaneous

Maps

```
> [...]          # load maps; not shown  
> map_plot(m)    # not shown
```

Output to Excel

```
> write.excel(m)  # not shown
```

Literature

- Harmening, Kreutzmann, Pannier, Skarke, Rojas-Perilla, Salvati, Schmid, Templ, Tzavidis & Würz (2021) emdi: Estimating and Mapping Disaggregated Indicators. R Package version 2.1.1. URL <https://CRAN.R-project.org/package=emdi>
- Kreutzmann, Pannier, Rojas-Perilla, Schmid, Templ & Tzavidis (2019) The R Package emdi for Estimating and Mapping Regionally Disaggregated Indicators, Journal of Statistical Software 91, p. 1-33. DOI 10.18637/jss.v091.i07
- Molina & Marhuenda (2020) sae: Small Area Estimation. R Package version 1.3, URL <https://CRAN.R-project.org/package=sae>
- Marhuenda, Morales & Pardo (2014) Information criteria for Fay-Herriot model selection. Computational Statistics and Data Analysis 70, p. 268-280.
- Molina & Marhuenda (2014) sae: An R Package for Small Area. The R Journal 7, p. 81-98.
- Schoch (2014) rsae: Robust Small Area Estimation. R Package version 0.1-5. URL <https://CRAN.R-project.org/package=rsae>