The tipsae Shiny app

Silvia De Nicolò
Università di Padova

Aldo Gardini
Università di Bologna

Abstract

The tipsae Shiny app is a dedicated R-based tool for mapping proportions and indicators defined on the unit interval, working in the framework of small area estimation. It implements Beta-based Bayesian Hierarchical models defined at area-level through Stan. A set of diagnostics, exploratory analysis and complementary tools complete the application.

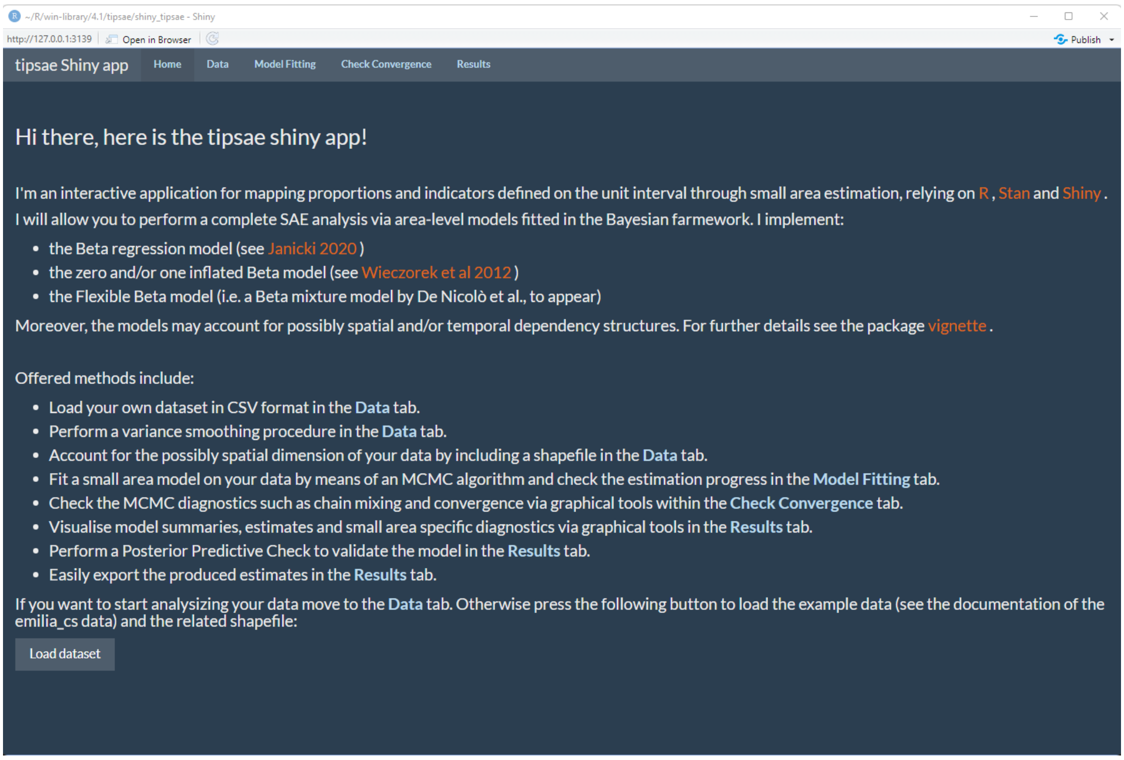
Keywords: Bayesian Inference, Beta Regression Models, Small Area Estimation, Shiny, Stan.

1. Introduction

Timely and reliable statistical estimates at a great level of disaggregation are increasingly in demand and require extensive exploitation of survey data. Nonetheless, domains or areas of study are often different from the ones for which the survey was originally planned, leading to unreliable survey estimates due to observations-poor and possibly non-representative samples. Small Area Estimation (SAE) techniques exploit auxiliary information to borrow strength across areas and produce estimates of interest with an acceptable level of uncertainty. Specifically, the area-level class of SAE models maps survey estimators of target quantities to areas-specific covariates, generally measured without error (e.g. census data), via an explicit regression model.

We focus on unit interval responses, common in SAE modelling because of the high presence of rates and proportions releases in official statistics. Two different bodies of literature relate with this field, revolving around linear mixed models with suitable transformations (Rao and Molina 2015) and Beta regression models (Janicki 2020). Although several routines for SAE have been released by developer teams in R, only the **emdi** package (Kreutzmann, Pannier, Rojas-Perilla, Schmid, Templ, and Tzavidis 2019) directly accounts for unit interval responses at area-level via Gaussian models with proper transformations. Furthermore, Beta-based small area models lack of proper implementation and the **tipsae** package, available on CRAN (De Nicolò and Gardini 2022), aims at filling this gap.

We implement area-level models based on the Beta likelihood comprising the standard Beta regression model, Zero and/or One Inflated Beta (Wieczorek, Nugent, and Hawala 2012) and Flexible Beta (De Nicolò, Ferrante, and Pacei 2021) models. Moreover, particular dependence structures can be modeled, including spatial and/or temporal random effects. We decided to operate in a Bayesian fashion via Stan routine (Carpenter, Gelman, Hoffman, Lee, Goodrich, Betancourt, Brubaker, Guo, Li, and Riddell 2017) to easily manage non-Gaussian assumptions, ease the treatment of the out-of-sample areas, and capture the uncertainty about



Hi there, here is the tipsae shiny app!

I'm an interactive application for mapping proportions and indicators defined on the unit interval through small area estimation, relying on R , Stan and Shiny .
I will allow you to perform a complete SAE analysis via area-level models fitted in the Bayesian famework. I implement:

- the Beta regression model (see Janicki 2020)
- the zero and/or one inflated Beta model (see Wieczorek et al 2012)
- the Flexible Beta model (i.e. a Beta mixture model by De Nicolò et al., to appear)

Moreover, the models may account for possibly spatial and/or temporal dependency structures. For further details see the package vignette .

Offered methods include:

- Load your own dataset in CSV format in the **Data** tab.
- Perform a variance smoothing procedure in the **Data** tab.
- Account for the possibly spatial dimension of your data by including a shapefile in the **Data** tab.
- Fit a small area model on your data by means of an MCMC algorithm and check the estimation progress in the **Model Fitting** tab.
- Check the MCMC diagnostics such as chain mixing and convergence via graphical tools within the **Check Convergence** tab.
- Visualise model summaries, estimates and small area specific diagnostics via graphical tools in the **Results** tab.
- Perform a Posterior Predictive Check to validate the model in the **Results** tab.
- Easily export the produced estimates in the **Results** tab.

If you want to start analysizing your data move to the **Data** tab. Otherwise press the following button to load the example data (see the documentation of the emilia_cs data) and the related shapefile:

Load dataset

Figure 1: Home page of the tipsae Shiny App.

target parameters through posterior inference. Within this framework, we developed a Shiny application to further facilitate the workflow for non-expert users.

Our application assists the user in carrying out a complete SAE analysis, starting from data loading, through the entire process of data exploration, model estimation and validation, presentation and exportation of results. This allows to straightforwardly use Bayesian models and complex SAE methods.

For the methodological details we refer to the package vignette “The R package tipsae: Tools for Mapping Proportions and Indicators on the Unit Interval”. In what follows, the workflow for a complete SAE analysis within the Shiny app is discussed. The application can be launched without any preliminary action by running the following commands.

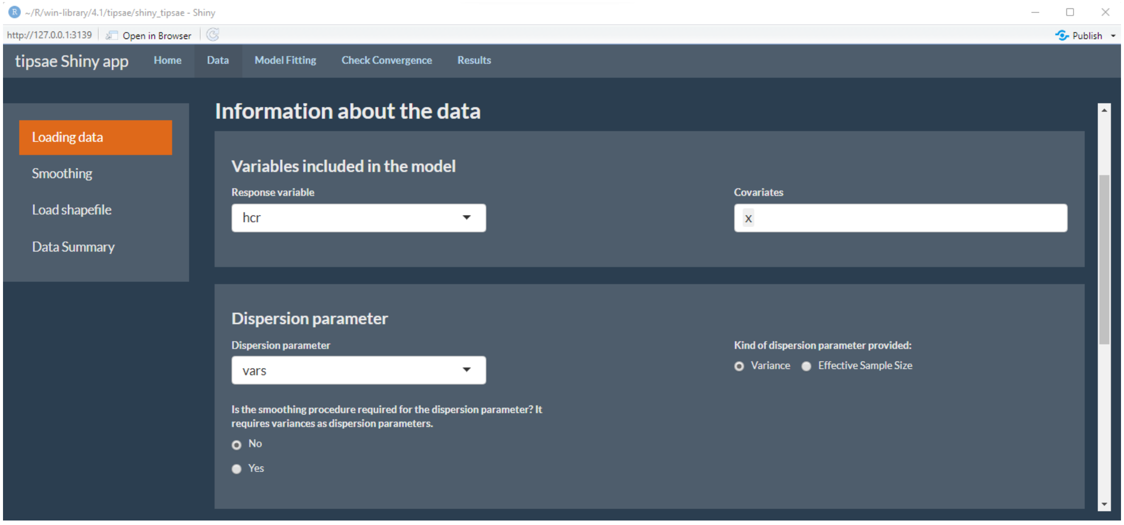
```
R> library(tipsae)
R> runShiny_tipsae()
```

A browser window pops up, allowing users to navigate the application.

The **Home** page (Figure 1) comprises a schematic description of the application with some relevant references. On the same page, the button “Load dataset" allows testing the application by performing an analysis through the emilia dataset, described in Section 3 of the package vignette.

The interface is organized into 4 main pages: the data input step is described in Section 2, the model fitting step and MCMC convergence checks are shown in Section 3 and results in Section 4.

2. Data Page: Input and Preliminary Analysis



Information about the data

Variables included in the model

Response variable: hcr

Covariates: x

Dispersion parameter

Dispersion parameter: vars

Kind of dispersion parameter provided:
○ Variance ● Effective Sample Size

Is the smoothing procedure required for the dispersion parameter? It requires variances as dispersion parameters.
○ No
● Yes

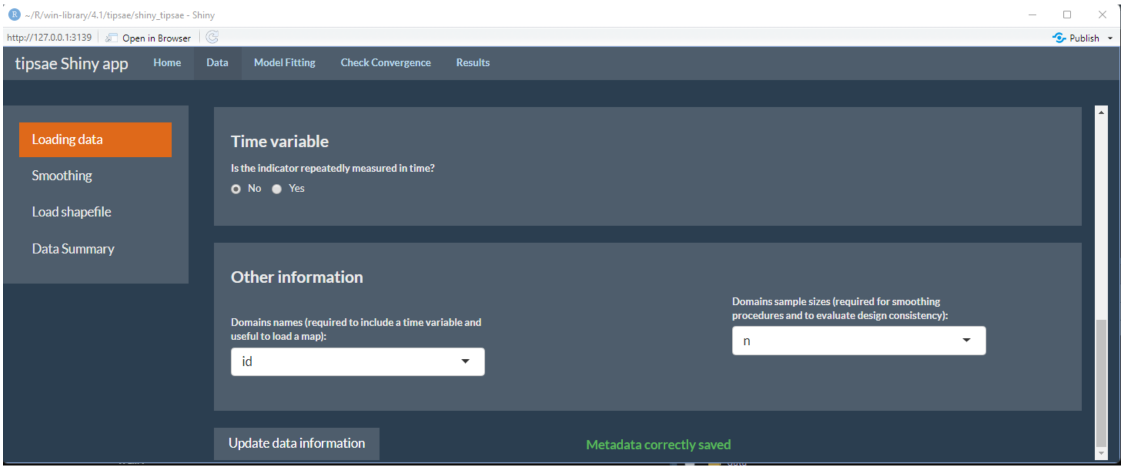
Figure 2: Home page of the tipsae Shiny App.

The **Data** page is divided into 4 subsections: the first 3 tabs concern data entry steps and data pre-treatment, while the latter provides graphical exploratory tools. In the *Loading Data* tab, a CSV file can be loaded and the data input may be visualized through the button "View loaded dataset". When the toy data are loaded on the **Home** page, the loading command does not appear. Additional information about data must be filled in, such as the nature of the inserted variables, the label of the response, covariates and dispersion values, specifying also if the latter is the variance or the effective sample size (Figure 2).

Here, it is also possible to set up the smoothing procedure, if needed: this possibility is available only when the variance is provided as input. Other information can be specified, such as a possible time variable (requested if multiple observations are present for the same areas), domain ids (mandatory for panel data) and sample sizes, useful for following visual diagnostics. To set the desired features to the loaded data, the “Update data information” button should be clicked, and if all the input variables are validated, a successful message appears (Figure 3).

When a smoothing procedure is set up, the *Smoothing* subsection enables it to change settings and graphically visualize its output. The procedure is a Generalized Variance Function smoothing technique (Fabrizi, Ferrante, Pacei, and Trivisano 2011); for a methodological explanation, refer to the package vignette. Note that only a simplified smoothing procedure is provided in the Shiny app if compared to the package function: it automatically assumes the response variable as a proportion, employing the related variance function. The user can choose between `"ols"` and `"gls"` regression types for smoothing.

In subsection *Load Shapefile*, a spatial structure can be incorporated, loading either an SHP file or an RDS file containing a `SpatialPolygonsDataFrame` object. Such an object would enable to account for spatial dependencies in the model and/or plot maps with relevant quantities. To match the shapefile and the estimates, the area-labels in the `SpatialPolygonsDataFrame` object should be consistent with the id variable of the loaded data.



Time variable

Is the indicator repeatedly measured in time?

● No ○ Yes

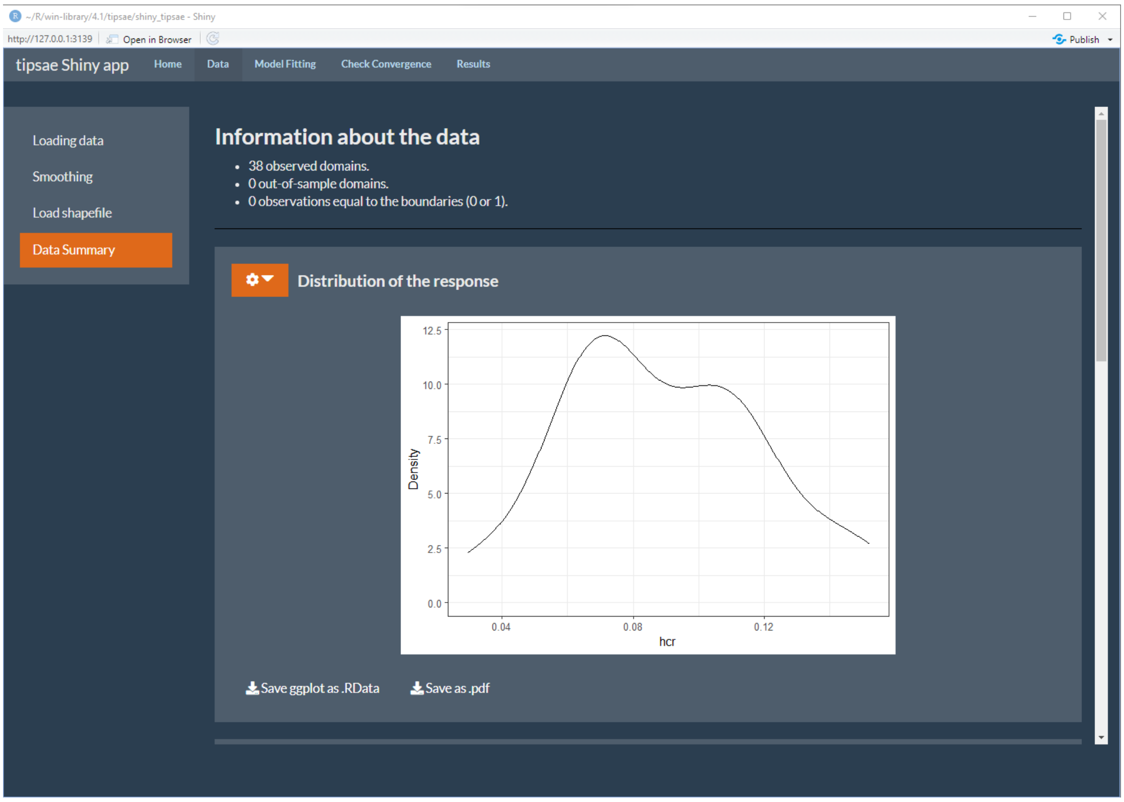
Other information

Domains names (required to include a time variable and useful to load a map): id

Domains sample sizes (required for smoothing procedures and to evaluate design consistency): n

Metadata correctly saved

Figure 3: *Loading data* tab of the tipsae Shiny App.

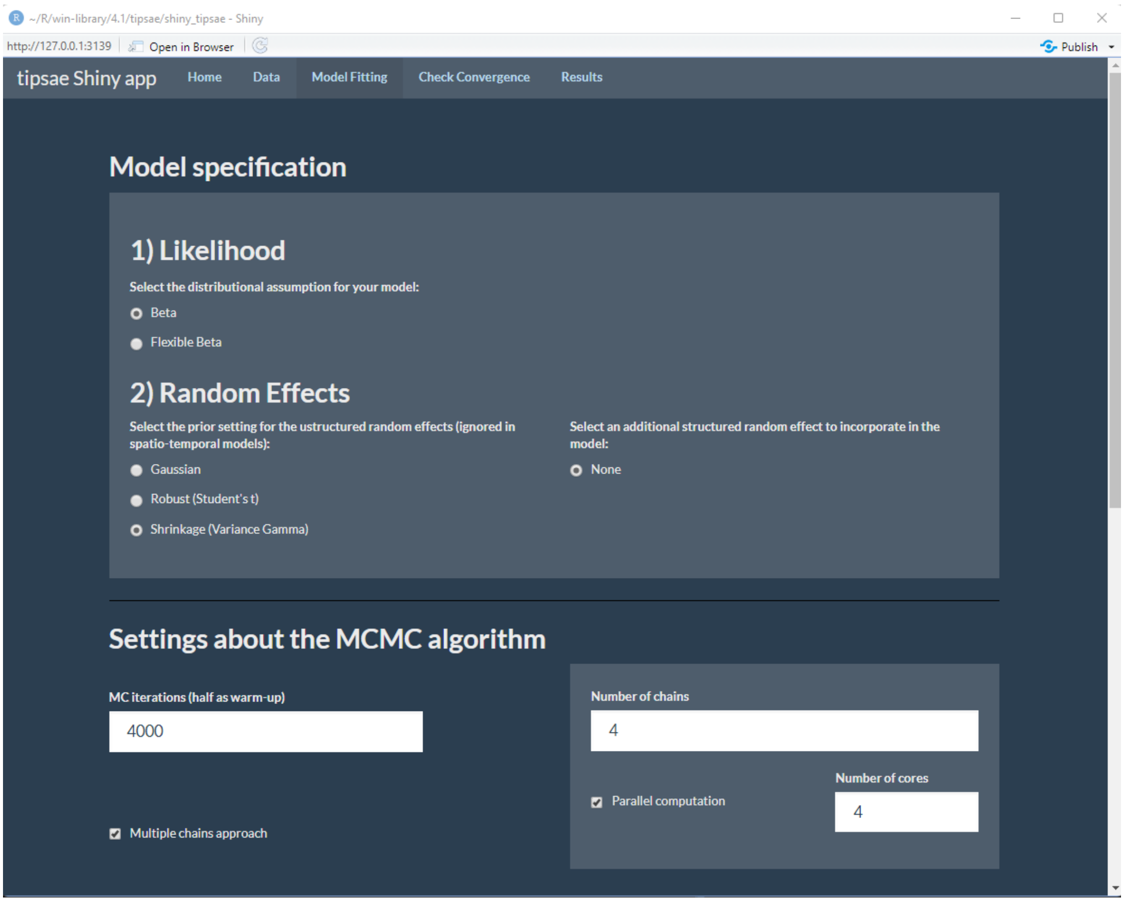


Information about the data

- 38 observed domains.
- 0 out-of-sample domains.
- 0 observations equal to the boundaries (0 or 1).

Distribution of the response

Figure 4: *Data summary* tab of the tipsae Shiny App.



Model specification

1) Likelihood

Select the distributional assumption for your model:

◉ Beta

● Flexible Beta

2) Random Effects

Select the prior setting for the ustructured random effects (ignored in spatio-temporal models):

● Gaussian

● Robust (Student's t)

◉ Shrinkage (Variance Gamma)

Select an additional structured random effect to incorporate in the model:

◉ None

Settings about the MCMC algorithm

MC iterations (half as warm-up): 4000

☑ Multiple chains approach

Number of chains: 4

☑ Parallel computation

Number of cores: 4

Figure 5: Model Fitting page of tipsae Shiny App: Model specification

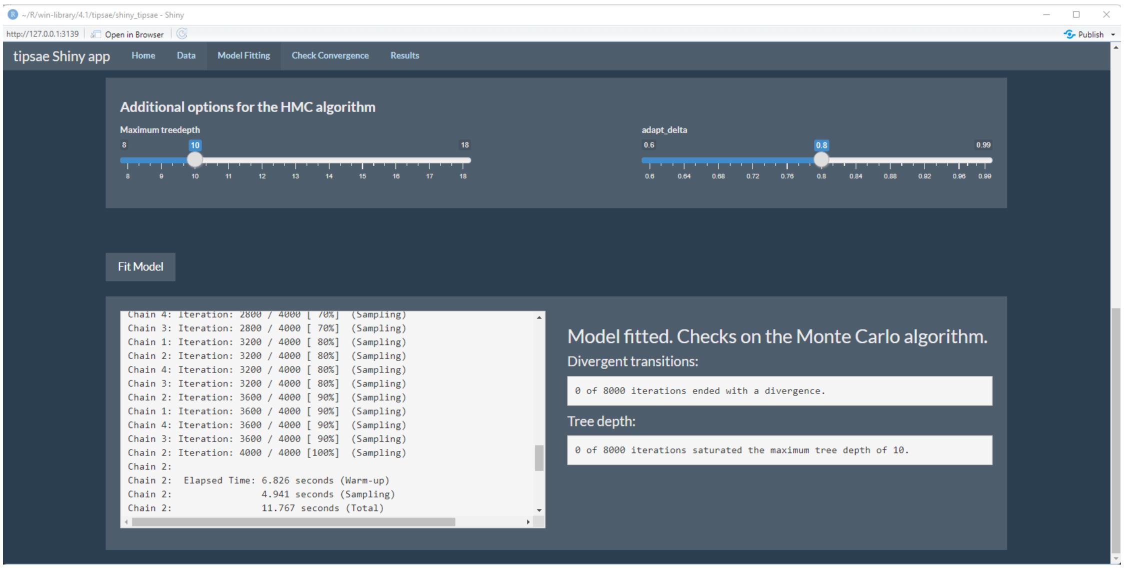
The last subsection, *Data Summary*, provides an accurate data exploration before moving to the modelling step, depicting the distribution of the response variable (Figure 4) and its relationship with the covariates and the dispersion measure. If the map is loaded, the spatial distribution of the variables involved in the analysis is shown too.

3. Model Fitting and Check Convergence

The **Model Fitting** page allows estimating Bayesian Beta-based small area models for indicators defined on the unit interval. The statistical features of such models are outlined in Section 3.1, whereas the details on the model specification and the estimation via Stan routine (Carpenter *et al.* 2017) are discussed in Section 3.2. Eventually, the **Check Convergence** page is deepened in Section 4.1.

3.1. Beta-based small area models

A classical Beta small area model for $y_d \in (0, 1)$, denoting the direct estimator of a generic target quantity and $\boldsymbol{x}_d$ being a set of P covariates for domain d, constitutes as a hierarchical model with two levels. The sampling level models the conditional distribution of the direct



Additional options for the HMC algorithm

Maximum treedepth: 10

adapt_delta: 0.8

```
Chain 4: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 3: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 1: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 2: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 4: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 3: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 2: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 1: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 4: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 3: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 2: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 2: 
Chain 2:  Elapsed Time: 6.826 seconds (Warm-up)
Chain 2:                4.941 seconds (Sampling)
Chain 2:                11.767 seconds (Total)
```

Model fitted. Checks on the Monte Carlo algorithm.

Divergent transitions:

```
0 of 8000 iterations ended with a divergence.
```

Tree depth:

```
0 of 8000 iterations saturated the maximum tree depth of 10.
```

Figure 6: Model Fitting Section of tipsae Shiny App

estimator as

$$y_d|\theta_d, \phi_d \overset{ind}{\sim} Beta(\theta_d\phi_d, (1-\theta_d)\phi_d), \quad \forall d.$$

where $\mathbb{E}[y_d|\theta_d, \phi_d] = \theta_d$ is the target parameter and is estimated via a logit regression at the linking level, i.e. $\text{logit}(\theta_d)|\boldsymbol{\beta}, v_d = \boldsymbol{x}_d^T\boldsymbol{\beta} + v_d$, with v_d being an area-specific random effect. Generally, a small area Beta model assumes the dispersion parameter ϕ_d as known in order to allow identifiability. In our package, we contemplate alternative likelihood assumptions at the sampling level:

- The Zero/One Inflated Beta model extends the support of y_d to zero/one values, by assuming a mixture of Beta and Dirac Delta components.
- The Flexible Beta distribution, defined as a mixture of two Beta components, allows to improve the modelling of target quantities with skewed and heavy-tailed distributions Migliorati, Di Brisco, and Ongaro (2018).

For additional details concerning the statistical models considered in the package, please refer to the package vignette.

3.2. Model Specification

The application allows exploiting a Stan-based efficient Hamiltonian Monte Carlo (HMC) fitting algorithm and customized parallel computing. In particular, the *Model Specification* tab (Figure 5) enables us to set the model likelihood: our application automatically constrains the model choice among those allowed by the input data. When any response observation $y_i \in (0, 1)$, $\forall i$ the allowed options are Beta and Flexible Beta models. If some observations are recorded as zeros and/or ones, the application automatically restricts the choice to the Zero and/or One Inflated Beta models. Secondly, the prior for the random effect can be

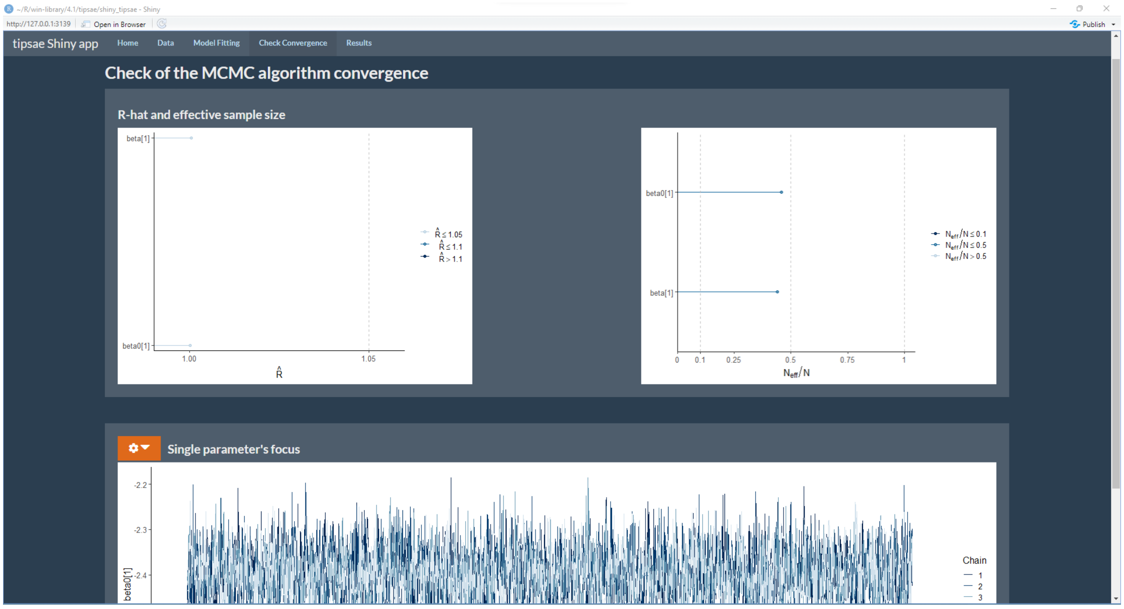


Figure 7: Convergence Assessment Section of tipsae Shiny App

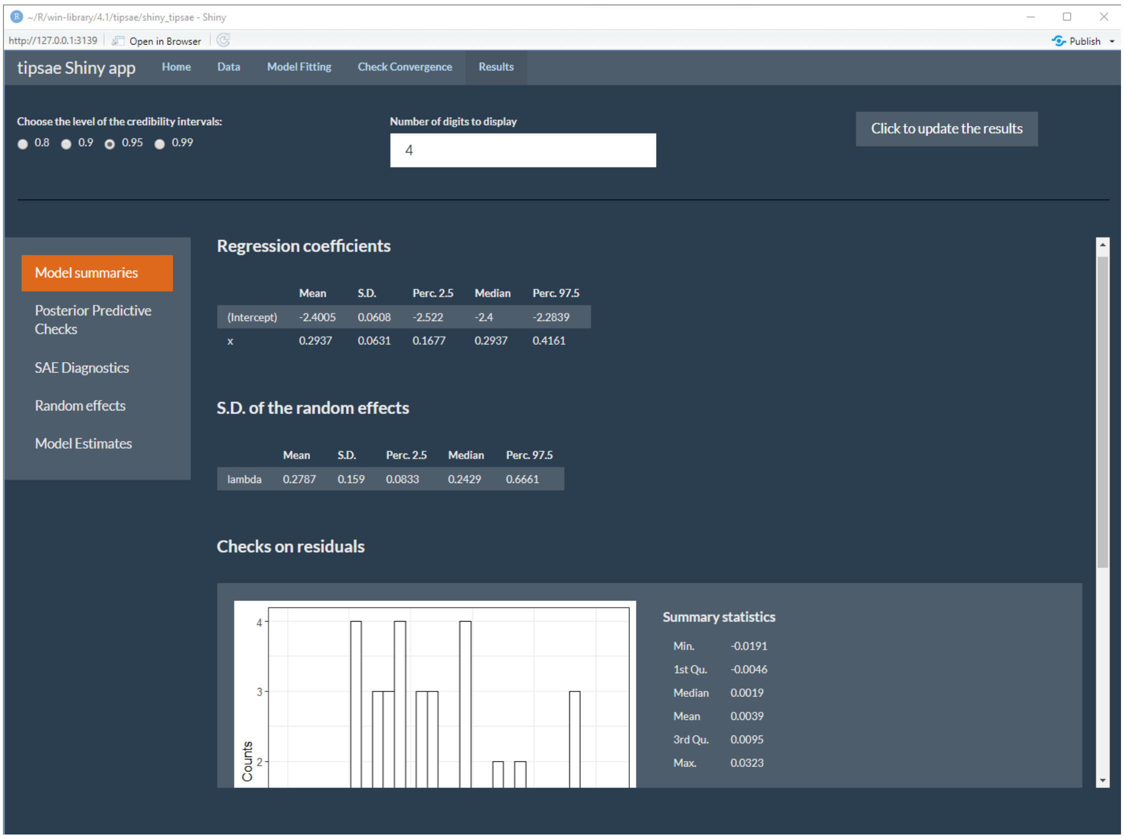
chosen among the classical Gaussian one, a robust alternative following a Student's t distribution (Fabrizi, Ferrante, and Trivisano 2016) and a shrinkage distribution, the variance gamma (Fabrizi, Ferrante, and Trivisano 2018), for a more flexible handling. When the data incorporates a time variable, therefore having a panel nature, the model automatically incorporates a temporal dependency structure, adding to the linear predictor an additional random effect with a random walk prior. Moreover, if a `SpatialPolygonsDataFrame` object related the to areas of interest have been loaded, the application asks whether to include a spatial dependency structure into the model or not. The dependency is incorporated via an additional random effect in the linear predictor with an intrinsic conditional autoregressive (ICAR) prior.

Some algorithm settings can also be modified in the *Settings about the MCMC Algorithm* tab, such as the number of iterations per chain, including warm-up (half of total iterations), the parallel computation to switch on with a proper tick, the number of chains, and the number of cores. Additional HMC options are the maximum allowed tree depth (*Maximum treedepth*) and the target average proposal acceptance probability (*adapt_delta*): refer to the Stan documentation for further details. The “Fit Model" button allows the user to start the estimation, whose progress is depicted by an iterative printed output (Figure 6).

4. Results

4.1. Convergence Assessment

Once computations are completed, the mixing of the MCMC algorithm can be checked through graphical tools within the **Check Convergence** page. In particular, for any linking level parameter, it is possible to visually inspect the posterior densities, chains trace-plots,



Choose the level of the credibility intervals:
● 0.8 ● 0.9 ○ 0.95 ● 0.99

Number of digits to display: 4

Regression coefficients

	Mean	S.D.	Perc. 2.5	Median	Perc. 97.5
(Intercept)	-2.4005	0.0608	-2.522	-2.4	-2.2839
x	0.2937	0.0631	0.1677	0.2937	0.4161

S.D. of the random effects

	Mean	S.D.	Perc. 2.5	Median	Perc. 97.5
lambda	0.2787	0.159	0.0833	0.2429	0.6661

Checks on residuals

Summary statistics

Min.	-0.0191
1st Qu.	-0.0046
Median	0.0019
Mean	0.0039
3rd Qu.	0.0095
Max.	0.0323

Figure 8: Results Section of tipsae Shiny App

autocorrelation functions and rank plots (Gabry and Mahr 2021). Additional useful information about the effectiveness of the HMC algorithm can be deduced from the plots of the $\hat{R}$ statistic (available if a multichain approach is chosen) and the effective sample size (Figure 7).

Eventually, the model results can be accessed in the **Results** page. The *Model Summaries* subsection provides posterior syntheses of regression coefficients and random effects. Furthermore, a summary of residuals is also reported, including a histogram, and the LOO Information Criterion can be computed through the button “Click to compute LOOIC", enabling for model selection. This criterion is based on approximate leave-one-out cross-validation computed using Pareto-smoothed importance sampling (Vehtari, Gelman, and Gabry 2017). The *Posterior Predictive Check* subsection displays the sample data kernel density versus those of the datasets generated from the posterior predictive distribution, denoted with $Y_d^\bullet|\boldsymbol{y}$, $d = 1, \ldots, D$, in order to assess the goodness of fit. Here, a specific tab focuses on area-specific Bayesian p-values, defined as $BP_d = \mathbb{P}\left[Y_d^\bullet > y_d|\boldsymbol{y}\right]$ (Fabrizi *et al.* 2011). In absence of systematic deviations, the expected Bayesian p-value is 0.5, whereas values near 0 or 1 highlight issues of over-estimation and under-estimation, respectively.

Small area-specific diagnostics have a proper subsection (*SAE diagnostics*), visually illustrating the shrinking process induced by the model and comparing direct versus model-based estimates. The standard deviations of both types of estimates are compared and summaries of a measure of standard deviation reduction are provided. The *Random Effect* subsection compares the densities of the standardized effects versus the one of a standard normal and a caterpillar plot, comparing their posterior distributions for each area. Lastly, the *Model Estimates* subsection displays a table with direct and model-based estimates, including relevant posterior summaries of target parameters. Such an object can be downloaded in CSV format via a proper button. A caterpillar plot of the target parameter posteriors is also provided.

5. Conclusion

The tipsae Shiny app ends up to be remarkable tool for small area mapping. Its ease of use let non-expert R users to carry out an analysis involving complex statistical methods under a Bayesian approach. This vignette is an attempt to describe the main points of the interface architecture.

References

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017). “Stan: A Probabilistic Programming Language.” *Journal of Statistical Software*, **76**(1), 1–32.

De Nicolò S, Ferrante MR, Pacei S (2021). “Put Inequality on the Map: Small Area Estimation of Inequality Measures using a Beta Mixture.” Working Paper.

De Nicolò S, Gardini A (2022). ***tipsae****: Tools for Handling Indices and Proportions in Small Area Estimation.* R package version 0.0.4, URL https://CRAN.R-project.org/package=tipsae.

Fabrizi E, Ferrante M, Trivisano C (2016). “Hierarchical Beta Regression Models for the Estimation of Poverty and Inequality Parameters in Small Areas.” *Analysis of Poverty Data by Small Area Methods. John Wiley and Sons*, pp. 299–314.

Fabrizi E, Ferrante MR, Pacei S, Trivisano C (2011). “Hierarchical Bayes Multivariate Estimation of Poverty Rates Based on Increasing Thresholds for Small Domains.” *Computational Statistics & Data Analysis*, **55**(4), 1736–1747.

Fabrizi E, Ferrante MR, Trivisano C (2018). “Bayesian Small Area Estimation for Skewed Business Survey Variables.” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **67**(4), 861–879.

Gabry J, Mahr T (2021). ***bayesplot****: Plotting for Bayesian Models.* R package version 1.8.1, URL https://mc-stan.org/bayesplot.

Janicki R (2020). “Properties of the Beta Regression Model for Small Area Estimation of Proportions and Application to Estimation of Poverty Rates.” *Communications in Statistics-Theory and Methods*, **49**(9), 2264–2284.

Kreutzmann AK, Pannier S, Rojas-Perilla N, Schmid T, Templ M, Tzavidis N (2019). “The R Package **emdi** for Estimating and Mapping Regionally Disaggregated Indicators.” *Journal of Statistical Software*, **91**(7), 1–33. doi:10.18637/jss.v091.i07.

Migliorati S, Di Brisco AM, Ongaro A (2018). “A New Regression Model for Bounded Responses.” *Bayesian Analysis*, **13**(3), 845–872.

Rao JN, Molina I (2015). *Small-Area Estimation*. Wiley Series in Survey Methodology.

Vehtari A, Gelman A, Gabry J (2017). “Practical Bayesian Model Evaluation Using Leave-One-Out Cross-Validation and WAIC.” *Statistics and Computing*, **27**(5), 1413–1432.

Wieczorek J, Nugent C, Hawala S (2012). “A Bayesian Zero-One Inflated Beta Model for Small Area Shrinkage Estimation.” In *Proceedings of the 2012 Joint Statistical Meetings, American Statistical Association, Alexandria, VA.*

Affiliation:

Silvia De Nicolò
Dipartimento di Scienze Statistiche
Università degli Studi di Padova
Via Cesare Battisti, 241
35121 Padova (PD), Italy
E-mail: silvia.denicolo@phd.unipd.it
URL: https://www.stat.unipd.it/ricerca/silvia-de-nicolo
and
Aldo Gardini
Dipartimento di Scienze Statistiche
Università di Bologna
Via Belle Arti, 41
40126, Bologna (BO), Italy
E-mail: aldo.gardini2@unibo.it
URL: https://www.unibo.it/sitoweb/aldo.gardini2/en