










sleev: An R Package for Semiparametric Likelihood Estimation with Errors in Variables

Jiangmei Xiong ^{1¶}, Sarah C. Lotspeich ², Joey B. Sherrill ³, Gustavo Amorim ¹, Bryan E. Shepherd ¹, and Ran Tao ^{1,4}

¹ Department of Biostatistics, Vanderbilt University Medical Center, USA ² Department of Statistical Sciences, Wake Forest University, USA ³ Brigham Young University, USA ⁴ Vanderbilt Genetics Institute, Vanderbilt University Medical Center, USA ¶ Corresponding author

DOI: [10.21105/joss.07320](https://doi.org/10.21105/joss.07320)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Mehmet Hakan Satman](#) 

Reviewers:

- [@alemermartinez](#)
- [@aalfons](#)

Submitted: 22 September 2024

Published: 01 August 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Data with measurement error in the outcome, covariates, or both are not uncommon, particularly with the increased use of routinely collected data for biomedical research. With error-prone data, often only a subsample of study data is validated; such settings are known as two-phase studies. The sieve maximum likelihood estimator (SMLE), which combines the error-prone data on all records with the validated data on a subsample, is a highly efficient and robust method to analyze such data. However, given their complexity, a computationally efficient and user-friendly tool is needed to obtain the SMLEs. The R package `sleev` fills this gap by making semiparametric likelihood-based inference using the SMLEs for error-prone two-phase data in settings with binary and continuous outcomes. Functions from this package can be used to analyze data with error-prone binary or continuous responses and error-prone covariates.

Statement of Need

Routinely collected data are being used frequently in biomedical research, such as electronic health records. However, these data tend to be error-prone, and using these data without correcting for their error-prone nature could lead to biased estimates and misleading research findings ([Duan et al., 2016](#)). To avoid such invalid study results, trained experts carefully verify and extract data elements. However, it is usually only feasible to validate data for a subset of records or variables. After validation, researchers have error-prone, pre-validation data for all records (phase one) and error-free validated data on a subset of records (phase two). Analyses aim to combine the two types of data to obtain estimates that have low bias and are as robust and efficient as possible.

There are several packages for R ([R Core Team, 2024](#)) that address measurement error, including `augSIMEX` ([Zhang & Yi, 2019](#)), `attenuation` ([Moss, 2019](#)), `decon` ([Wang & Wang, 2011](#)), `eivtools` ([Lockwood, 2018](#)), `GLSME` ([Hansen & Bartoszek, 2012](#)), `mecor` ([Nab et al., 2021](#)), `meerva` ([Kremers, 2021](#)), `mmc` ([Song, 2015](#)), `refitME` ([Stoklosa et al., 2021](#)), and `simex` ([Lederer & Seibold, 2019](#)). The various R packages reflect many different approaches, such as regression calibration ([Wang & Wang, 2011](#)), SIMEX (i.e., simulation-extrapolation) ([Lederer & Seibold, 2019](#)), and moment-based corrections ([Nab et al., 2021](#)), to mention a few. Nearly all of these existing R packages deal with errors in either the outcome or covariates, but not both, and none of these packages permits efficient inference that incorporates both the error-prone phase-one data and the validated phase-two data.

The sieve maximum likelihood estimator (SMLE) is an estimator that analyzes two-phase data by combining the error-prone data on all records with the validated data on a subsample. By leveraging all available data, the SMLE operates with high efficiency ([Lotspeich et al., 2022](#);

Tao et al., 2021). Since it does not make any parametric assumptions on the error model, the SMLE is also robust. For example, Tao et al. (2021) performed a set of simulations highlighting the SMLE's robustness to different error mechanisms including settings where the errors had non-zero mean or were multiplicative. Moreover, the SMLE allows error-prone outcome and error-prone covariates in the same model. Still, in practice these estimators can be difficult to implement, as they involve approximating nuisance conditional densities using B-splines (Schumaker, 2007) and then maximizing the semiparametric likelihood via a sophisticated EM algorithm (Tao et al., 2017). Here, we present the R package `sleev`, which makes the SMLE readily applicable for practitioners in a user-friendly way. `sleev` integrates and extends primitive R packages, `logreg2ph` and `TwoPhaseReg`, developed with the original methods papers (Lotspeich et al., 2022; Tao et al., 2021). These two packages lacked proper documentation and were difficult to use. `logreg2ph` was also computationally slow.

To promote the use of the SMLE, extensive work has been done to create `sleev`, a computationally efficient and user-friendly R package to analyze two-phase, error-prone data. Specifically, in `sleev` we rewrote the core algorithms of `logreg2ph` in C++ to speed up the computation, and we unified the syntax across functions. To compare the computational times, we set up simulations with the same code in the [package vignette](#). The simulations included phase-one and phase-two sample sizes of 2087 and 835, respectively, and were performed on a 64-bit Linux OS machine with 8G memory. Across 100 simulations, the previous `logreg2ph` took an average of 289.44 seconds with a standard deviation of 8.83 seconds to perform the analysis, while the corresponding new function in `sleev` only took an average of 122.32 seconds with a standard deviation of 8.18 seconds.

SMLE for Linear Regression

In this section, we briefly introduce the SMLE for linear regression. Suppose that we want to fit a standard linear regression model for a continuous outcome Y and covariates \mathbf{X} : $Y = \alpha + \beta^T \mathbf{X} + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$. Our goal is to obtain estimates of $\boldsymbol{\theta} = (\alpha, \beta^T, \sigma^2)^T$. When we have error-prone data, Y and \mathbf{X} are unobserved except for a subset of validated records. For unvalidated records (the majority), only the error-prone outcome $Y^* = Y + W$ and covariates $\mathbf{X}^* = \mathbf{X} + \mathbf{U}$ are observed in place of Y and \mathbf{X} , where W and \mathbf{U} are the errors for the outcome and covariates, respectively. We assume that W and \mathbf{U} are independent of ϵ . With potential errors in our data, a naive regression analysis using error-prone variables Y^* and \mathbf{X}^* could render misleading results (Fuller, 2009).

We assume that the joint density of the complete data $(Y^*, \mathbf{X}^*, W, \mathbf{U})$ takes the form

$$\begin{aligned} P(Y^*, \mathbf{X}^*, W, \mathbf{U}) &= P(Y^* | \mathbf{X}^*, W, \mathbf{U}) P(W, \mathbf{U} | \mathbf{X}^*) P(\mathbf{X}^*) \\ &= P_{\boldsymbol{\theta}}(Y | \mathbf{X}) P(W, \mathbf{U} | \mathbf{X}^*) P(\mathbf{X}^*), \end{aligned}$$

where $P(\cdot)$ and $P(\cdot | \cdot)$ denote density and conditional density functions, respectively. Specifically, $P_{\boldsymbol{\theta}}(Y | \mathbf{X})$ then refers to the conditional density function of the linear regression model of Y given \mathbf{X} . Denote the validation indicator variable by V , with $V = 1$ indicating that a record was validated and $V = 0$ otherwise. For records with $V = 0$, their measurement errors (W, \mathbf{U}) are missing, and therefore their contributions to the log-likelihood can be obtained by integrating out W and \mathbf{U} .

Let $(Y_i^*, \mathbf{X}_i^*, W_i, \mathbf{U}_i, V_i, Y_i, \mathbf{X}_i)$ for $i = 1, \dots, n$ denote independent and identically distributed realizations of $(Y^*, \mathbf{X}^*, W, \mathbf{U}, V, Y, \mathbf{X})$ in a sample of n subjects. Then, the observed-data log-likelihood is proportional to

$$\sum_{i=1}^n V_i \{ \log P_{\theta}(Y_i | \mathbf{X}_i) + \log P(W_i, \mathbf{U}_i | \mathbf{X}_i^*) \} + \sum_{i=1}^n (1 - V_i) \log \left\{ \int \int P_{\theta}(Y_i^* - w | \mathbf{X}_i^* - \mathbf{u}) P(w, \mathbf{u} | \mathbf{X}_i^*) dw d\mathbf{u} \right\}, \quad (1)$$

where $P(\mathbf{X}^*)$ is left out, because the error-prone covariates are fully observed and thus $P(\mathbf{X}^*)$ can simply be estimated empirically. We estimate the unknown measurement error model, $P(W_i, \mathbf{U}_i | \mathbf{X}_i^*)$, using B-spline sieves. Specifically, we approximate $P(w, \mathbf{u} | \mathbf{X}_i^*)$ and $\log P(W_i, \mathbf{U}_i | \mathbf{X}_i^*)$ by $\sum_{k=1}^m \mathbf{I}(w = w_k, \mathbf{u} = \mathbf{u}_k) \sum_{j=1}^{s_n} B_j^q(\mathbf{X}_i^*) p_{kj}$ and $\sum_{k=1}^m \mathbf{I}(W_i = w_k, \mathbf{U}_i = \mathbf{u}_k) \sum_{j=1}^{s_n} B_j^q(\mathbf{X}_i^*) \log p_{kj}$, respectively. Here, $\{(w_1, \mathbf{u}_1), \dots, (w_m, \mathbf{u}_m)\}$ are the m distinct observed (W, \mathbf{U}) values from the validation study, $B_j^q(\mathbf{X}_i^*)$ is the j th B-spline basis function of order q evaluated at \mathbf{X}_i^* , s_n is the dimension of the B-spline basis, and p_{kj} is the coefficient associated with $B_j^q(\mathbf{X}_i^*)$ and (w_k, \mathbf{u}_k) . The expression (1) is now approximated by

$$\sum_{i=1}^n V_i \left[\log P_{\theta}(Y_i | \mathbf{X}_i) + \sum_{k=1}^m \left\{ \mathbf{I}(W_i = w_k, \mathbf{U}_i = \mathbf{u}_k) \sum_{j=1}^{s_n} B_j^q(\mathbf{X}_i^*) \log p_{kj} \right\} \right] + \sum_{i=1}^n \log \left[\sum_{k=1}^m \left\{ P_{\theta}(Y_i^* - w_k | \mathbf{X}_i^* - \mathbf{u}_k) \sum_{j=1}^{s_n} B_j^q(\mathbf{X}_i^*) \log p_{kj} \right\} \right]. \quad (2)$$

The maximization of expression (2) is carried out through an EM algorithm to find the SMLEs $\hat{\theta}$ and \hat{p}_{kj} . The covariance matrix of the SMLE $\hat{\theta}$ is obtained through the method of profile likelihood (Murphy & Van der Vaart, 2000).

The SMLEs for logistic regression are similar to linear regression and described in the [package vignette](#), and the theoretical properties can be found in Lotspeich et al. (2022).

Functionalities of the `sleev` R Package

The `sleev` package provides a user-friendly way to obtain the SMLEs and their standard errors. The package can be installed from [CRAN](#) or [GitHub](#). The `sleev` package includes two main functions: `linear2ph()` and `logistic2ph()`, to fit linear and logistic regressions, respectively, under two-phase sampling with an error-prone outcome and covariates. The input arguments are similar for the two functions and listed in Table 1. In addition to the arguments for error-prone and error-free outcome and covariates, the user needs to specify the B-spline matrix $B_j^q(\mathbf{X}_i^*)$ to be used in the estimation of the error densities.

Table 1: Main arguments for the `linear2ph()` and `logistic2ph()` functions

Argument	Description
<code>y_unval</code>	Column name of unvalidated outcome in the input dataset.
<code>y</code>	Column name of validated outcome in the input dataset. NAs in the input will be counted as individuals not selected in phase two.
<code>x_unval</code>	Column names of unvalidated covariates in the input dataset.
<code>x</code>	Column names of validated covariates in the input dataset. NAs in the input will be counted as individuals not selected in phase two.
<code>z</code>	Column names of error-free covariates in the input dataset.
<code>data</code>	Dataset generated from <code>sleev::splines2ph()</code> .

Argument	Description
hn_scale	Scale of the perturbation constant in the variance estimation via the method of profile likelihood. The default is 1.
se	Standard errors of the parameter estimators will be estimated when set to TRUE. The default is TRUE.
tol	Convergence criterion. The default is 0.0001.
max_iter	Maximum number of iterations in the EM algorithm. The default is 1000.
verbose	Print analysis details when set to TRUE. The default is FALSE.

Example: Case study with mock data

For demonstration, the `sleev` package includes a dataset constructed to mimic data from the Vanderbilt Comprehensive Care Clinic (VCCC) patient records from Giganti et al. (2020). Table 2 describes the variables in this dataset.

Table 2: Data dictionary for `mock.vccc`

Name	Status	Type	Description
ID	error-free		Patient ID
VL_unval	error-prone	continuous	Viral load (VL) at antiretroviral therapy (ART) initiation
VL_val	validated	continuous	
ADE_unval	error-prone	binary	Had an AIDS-defining event (ADE) within one year of ART initiation: 1 - yes, 0 - no
ADE_val	validated	binary	
CD4_unval	error-prone	continuous	CD4 count at ART initiation
CD4_val	validated	continuous	
Prior_ART	error-free	binary	Experienced ART before enrollment: 1 - yes, 0 - no
Sex	error-free	binary	Sex at birth of patient: 1 - male, 0 - female
Age	error-free	continuous	Age of patient

We now illustrate how to obtain the SMLEs using the `sleev` package with the `mock.vccc` dataset. Specifically, we show how to fit a linear regression model in the presence of errors in both the outcome and covariates using the `linear2ph()` function. Situations with more covariates and examples with logistic regression are included in the [package vignette](#).

This example fits a linear regression model with CD4 count at antiretroviral therapy (ART) initiation regressed on viral load (VL) at ART initiation, adjusting for sex at birth. Both CD4 and VL are error-prone, partially validated variables, whereas sex is error-free. Because of skewness, we often transform both CD4 and VL. In our analysis, CD4 was divided by 10 and square root transformed, and VL was \log_{10} transformed:

```
library("sleev")
data("mock.vccc")
mock.vccc$CD4_val_sq10 <- sqrt(mock.vccc$CD4_val / 10)
mock.vccc$CD4_unval_sq10 <- sqrt(mock.vccc$CD4_unval / 10)
mock.vccc$VL_val_l10 <- log10(mock.vccc$VL_val)
mock.vccc$VL_unval_l10 <- log10(mock.vccc$VL_unval)
```

To obtain the SMLEs, we first need to set up the B-spline basis for the error-prone covariate `VL_unval_l10` (the transformed VL variable from phase one) and Sex. The `spline2ph()` function in the `sleev` package can set up the B-spline basis, and combine it with the input data for the final analysis. Here, we use a cubic B-spline basis with the `degree = 3` argument. The size of the basis s_n is set to be 20, specified through the `size = 20` argument. More

details regarding order and size selection, as well as run time comparison of B-spline basis, are discussed in the [package vignette](#). To allow possible heterogeneity in error distribution between males and females, we can set up B-spline basis separately and proportionally for the two Sex groups by specifying argument `group = "Sex"`. The described B-spline basis is constructed as follows.

```
sn <- 20
data.linear <- spline2ph(x = "VL_unval_l10", data = mock.vccc, size = sn,
                        degree = 3, group = "Sex")
```

Alternatively, if the investigator has prior knowledge that the errors in `VL_unval_l10` are likely to be homogeneous, one may fit a simpler model by not stratifying the B-spline basis by Sex.

Having constructed the B-spline basis, the SMLEs can be obtained by running the `linear2ph()` function on `data.linear`, as shown in the code below. Again, the inputs are explained in Table 1. The fitted SMLEs are stored in a list object of class `linear2ph`. Here, we assign the fitted SMLEs to the variable name `res_linear`. The list of class `linear2ph` contains five components: `coefficient`, `covariance`, `sigma`, `converge`, and `converge_cov`.

```
res_linear <- linear2ph(y_unval = "CD4_unval_sq10", y = "CD4_val_sq10",
                      x_unval = "VL_unval_l10", x = "VL_val_l10",
                      z = "Sex", data = data.linear,
                      hn_scale = 1, se = TRUE, tol = 1e-04,
                      max_iter = 1000, verbose = FALSE)
```

We should first check if the EM algorithms for estimating the regression coefficients and their covariance matrix converged by using the `print()` for class `linear2ph` directly.

```
> res_linear
```

Call:

```
linear2ph(y_unval = "CD4_unval_sq10", y = "CD4_val_sq10",
          x_unval = "VL_unval_l10", x = "VL_val_l10", z = "Sex",
          data = data.linear, hn_scale = 1, se = TRUE,
          tol = 1e-04, max_iter = 1000, verbose = FALSE)
```

The parameter estimation has converged.

Coefficients:

```
Intercept VL_val_l10      Sex
4.8209166 -0.1413168  0.2727984
```

The `summary()` function for the object of class `linear2ph` returns the estimated coefficients, their standard errors, test statistics, and *p*-values as follows:

```
> summary(res_linear)
```

Call:

```
linear2ph(y_unval = "CD4_unval_sq10", y = "CD4_val_sq10",
          x_unval = "VL_unval_l10", x = "VL_val_l10", z = "Sex",
          data = data.linear, hn_scale = 1, se = TRUE,
          tol = 1e-04, max_iter = 1000, verbose = FALSE)
```

Coefficients:

	Estimate	SE	Statistic	p-value
Intercept	4.8209166	0.15865204	30.386729	0.0000000000
VL_val_l10	-0.1413168	0.03983406	-3.547636	0.0003887047
Sex	0.2727984	0.10888178	2.505455	0.0122294098

Acknowledgement

This research was supported by the National Institute of Health grants R01AI131771, R01HL094786, and P30AI110527 and the 2022 Biostatistics Faculty Development Award from the Department of Biostatistics at Vanderbilt University Medical Center. This work leveraged the resources provided by the Vanderbilt Advanced Computing Center for Research and Education (ACCRES), a collaborative operated by and for Vanderbilt faculty.

References

- Duan, R., Cao, M., Wu, Y., Huang, J., Denny, J. C., Xu, H., & Chen, Y. (2016). An empirical study for impacts of measurement errors on EHR based association studies. *AMIA Annual Symposium Proceedings, 2016*, 1764.
- Fuller, W. A. (2009). *Measurement error models*. John Wiley & Sons. <https://doi.org/10.1002/9780470316665>
- Giganti, M. J., Shaw, P. A., Chen, G., Bebawy, S. S., Turner, M. M., Sterling, T. R., & Shepherd, B. E. (2020). Accounting for dependent errors in predictors and time-to-event outcomes using electronic health records, validation samples, and multiple imputation. *The Annals of Applied Statistics*, 14(2), 1045. <https://doi.org/10.1214/20-aos1343>
- Hansen, T. F., & Bartoszek, K. (2012). Interpreting the evolutionary regression: The interplay between observational and biological errors in phylogenetic comparative studies. *Systematic Biology*, 61(3), 413–425. <https://doi.org/10.1093/sysbio/syr122>
- Kremers, W. K. (2021). *meerva: Analysis of data with measurement error using a validation subsample*. <https://doi.org/10.32614/CRAN.package.meerva>
- Lederer, W., & Seibold, H. (2019). *Simex: SIMEX- and MCSIMEX-algorithm for measurement error models*. <https://doi.org/10.32614/CRAN.package.simex>
- Lockwood, J. R. (2018). *eivtools: Measurement error modeling tools*. <https://doi.org/10.32614/CRAN.package.eivtools>
- Lotspeich, S. C., Shepherd, B. E., Amorim, G. G., Shaw, P. A., & Tao, R. (2022). Efficient odds ratio estimation under two-phase sampling using error-prone data from a multi-national HIV research cohort. *Biometrics*, 78(4), 1674–1685. <https://doi.org/10.1111/biom.13512>
- Moss, J. (2019). *Attenuation: Correcting for attenuation due to measurement error*. <https://doi.org/10.32614/CRAN.package.attenuation>
- Murphy, S. A., & Van der Vaart, A. W. (2000). On profile likelihood. *Journal of the American Statistical Association*, 95(450), 449–465. <https://doi.org/10.2307/2669386>
- Nab, L., Smeden, M. van, Keogh, R. H., & Groenwold, R. H. (2021). Mecor: An R package for measurement error correction in linear regression models with a continuous outcome. *Computer Methods and Programs in Biomedicine*, 208, 106238. <https://doi.org/10.1016/j.cmpb.2021.106238>
- R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://doi.org/10.32614/r.manuals>
- Schumaker, L. (2007). *Spline functions: Basic theory*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511618994>
- Song, J. (2015). *mmc: Multivariate measurement error correction*. <https://doi.org/10.32614/CRAN.package.mmc>
- Stoklosa, J., Hwang, W., & Warton, D. (2021). *refitME: Measurement error modelling using MCEM*. <https://doi.org/10.32614/CRAN.package.refitME>

- Tao, R., Lotspeich, S. C., Amorim, G., Shaw, P. A., & Shepherd, B. E. (2021). Efficient semiparametric inference for two-phase studies with outcome and covariate measurement errors. *Statistics in Medicine*, 40(3), 725–738. <https://doi.org/10.1002/sim.8799>
- Tao, R., Zeng, D., & Lin, D. Y. (2017). Efficient semiparametric inference under two-phase sampling, with applications to genetic association studies. *Journal of the American Statistical Association*, 112(520), 1468–1476. <https://doi.org/10.1080/01621459.2017.1295864>
- Wang, X. F., & Wang, B. (2011). Deconvolution estimation in measurement error models: The R package decon. *Journal of Statistical Software*, 39, 1–24. <https://doi.org/10.18637/jss.v039.i10>
- Zhang, Q., & Yi, G. Y. (2019). R package for analysis of data with mixed measurement error and misclassification in covariates: augSIMEX. *Journal of Statistical Computation and Simulation*, 89(12), 2293–2315. <https://doi.org/10.1080/00949655.2019.1615911>