# Package: winputall (via r-universe)

January 11, 2025

**Title** Variable Input Allocation Among Crops

**Version** 1.0.1

**Description** Using a time-varying random parameters model developed in Koutchade et al., (2024) <https://hal.science/hal-04318163>, this package allows allocating variable input costs among crops produced by farmers based on panel data including information on input expenditure aggregated at the farm level and acreage shares. It also considers in fairly way the weighting data and can allow integrating time-varying and time-constant control variables.

**License** GPL (>= 3)

**Depends** R (>= 3.4.0)

**Imports** dplyr, future, future.apply, graphics, ks, LearnBayes, MASS, Matrix, matrixcalc, matrixStats, methods, plm, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.4.0), stats, utils

**Suggests** spelling, testthat (>= 3.0.0)

**LinkingTo** BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

**Biarch** true

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.1

**SystemRequirements** GNU make

**Language** en-US

**NeedsCompilation** yes

**Author** Obafèmi Philippe Koutchade [aut, cre, cph] (<https://orcid.org/0000-0001-7327-3139>), Fabienne Femenia [aut], Alain Carpentier [aut]

**Maintainer** Obafèmi Philippe Koutchade

<obafemi-philippe.koutchade@inrae.fr>

**Date/Publication** 2024-06-07 14:40:02 UTC

**Config/pak/sysreqs** make

**Repository** https://opkoutchade.r-universe.dev

**RemoteUrl** https://github.com/cran/winputall

**RemoteRef** HEAD

**RemoteSha** aeac2a473bd540f07fede9270c646dbc7608eaca

# Contents



---

`winputall-package` *The 'winputall' package.*

---

## Description

Using a time-varying random parameters model, this package allows allocating variable input costs among crops produced by farmers based on panel data including information on input expenditure aggregated at the farm level and acreage shares. It also considers in fairly way the weighting data and can allow integrating time-varying and time-constant control variables.

## Author(s)

**Maintainer**: Obafèmi Philippe Koutchade <obafemi-philippe.koutchade@inrae.fr> (ORCID) [copyright holder]

Authors:

- Fabienne Femenia <fabienne.femenia@inrae.fr>
- Alain Carpentier <alain.carpentier@inrae.fr>

## References

Koutchade Obafèmi Philippe, Fabienne Femenia, Alain Carpentier (2024).Variable Input Allocation Among Crops: A Time-Varying Random Parameters Approach https://hal.science/hal-04318163

## my_winputall_data *Example data*

### Description

An unbalanced panel data

### Usage

```
my_winputall_data
```

### Format

A data frame with 175 rows and 9 variables:

**id** individual index

**year** time index

**tx** Total input uses

**s_crop1** acreage of crop number 1

**s_crop2** acreage of crop number 2

**s_crop3** acreage of crop number 3

**k** Capital

**tempmoy** Temperature

**precip** Annual precipitation ...

## print.rpinpall *Fit Input Allocation Random Parameters Model*

### Description

Designed to fit a random parameters input allocation model proposed in Koutchade et al., (2024) https://hal.science/hal-04318163. It provides crops input cost for each individual at each time and can account for Weighted Panel Data.

### Usage

```
## S3 method for class 'rpinpall'
print(x, error = FALSE, ...)

## S3 method for class 'rpinpall'
summary(object, ...)

## S3 method for class 'rpinpall'
```

```
plot(x, ...)

rpinpallEst(
  data,
  id_time,
  total_input,
  crop_acreage,
  crop_indvar = NULL,
  crop_rp_indvar = NULL,
  weight = NULL,
  distrib_method = c("lognormal", "normal", "censored-normal"),
  sim_method = c("map_imh", "mhrw", "marg_imh", "mhrw_imh", "nuts", "variat",
    "lapl_approx"),
  calib_method = c("cmode", "cmean", "rscd", "estim-sim"),
  saem_control = list(),
  par_init = list()
)
```

## Arguments

| | |
|---|---|
| x | An object produced by the function rpinpallEst, to be displayed |
| error | logical. If TRUE, residuals are considered in variable input prediction |
| ... | Other arguments |
| object | An object produced by the function rpinpallEst, to be displayed |
| data | name of the data frame or matrix containing all the variables included in the model. |
| id_time | first (individual) and second (time) level variables allowing characterizing panel data. |
| total_input | variable (name) containing the total input used at farm level per ha to be allocated to the different crops. |
| crop_acreage | list of variables containing the acreage of the different crops. |
| crop_indvar | optional list of vector of (time-varying) variables specific to each crop used to control for observed (individual and/or temporal) characteristics in the estimation process. Default=NULL. |
| crop_rp_indvar | optional list of vector of (time-constant) variables specific to each crop used to control for observed time-constant characteristics in the estimation process. Default=NULL. |
| weight | optional variable containing weights of individual sample farms. Default=NULL (equal weight is given to each farm). Default=NULL. |
| distrib_method | assumption on the distribution of input use per crop (x_kit): "normal", "lognormal" or "censored-normal". Default="lognormal". |
| sim_method | method used to draw the random parameters in the simulation step of the SAEM algorithm in the estimation process: "map_imh" (independant Metropolis Hasting with Laplace approximation as proposal distribution), "marg_imh" (independant Metropolis Hasting with marginal distribution of random pararameters as |

| | |
|---|---|
| | proposal distribution), "mhrw" (Metropolis Hasting Random Walk), "mhrw_imh" (combined "imh" and "mhrw"), "nuts","variat" and"lapl_approx". Default= "map_imh". |
| calib_method | method used: "cmode" (conditional mode), "cmean" (conditional mean). Default="cmode". |
| saem_control | list of options for the SAEM algorithm. See 'Details |
| par_init | list of some parameters' initialization. |

## Details

An SAEM algorithm is used to perform the estimation of input uses per crop. Different options can be specified by the user for this algorithm in the saem_control argument. The saem_control argument is list that can supply any of the following component.

- nb_burn_saem: Number of iterations of the burn-in phase where individual parameters are sampled from their conditional distribution using sim_method and the initial values for model parameters without update these parameters. Default=20.
- nb_SA: Number of iterations in the exploration phase where algorithm explore parameters space without memory. The parameter that controls the convergence of the algorithm is set to 1. Default=200.
- nb_smooth: Number of iterations in the smoothing phase.Default=200 and the parameter that controls the convergence of the algorithm is set to 0.85 by default.
- nb_RS: Number of iterations where tempering approach is used
- tol: Tolerance value for the convergence. Default 1.10-3.
- estim_rdraw: Number of random draws using in the estimation process. Default=100
- calib_rdraw: Number of random draws using in the calibration process. Default=100
- stde_rdraw: Number of random draws using for computation of estimation standard errors. Default=100
- p_SA: Parameter determining step sizes in the Stochastic Approximation (SA) step. Must be comprise between 0 and 1. Default=0.85
- doParallels: Logical.If TRUE a parallel processing is used when more than 2 cores are available. Default=FALSE
- doTempering: Logical. If TRUE the tempering approach proposed by (Allassonnière and Chevallier, 2021) is used to avoid convergence to local maxima. Default=TRUE
- doDiagEps: = "2",
- showProgress: Logical. If TRUE the evolution of the estimation process is displayed graphically at the bottom of the screen. Default=TRUE
- showIterConvLL: Logical. If TRUE iteration number and convergence value are displayed during the estimation process. Default=FALSE

## Value

Distribution of estimated crop input uses.

This function returns a list with the following components:

- xit_pred: matrix of predicted crop input used per ha.

- `xit_pred_with_error`: matrix of predicted crop input used per ha.
- `yit_predict`: vector of predicted totat input used.
- `est_pop` list of results of estimation: estimated parameters.
- `est_stdelist` of parameters standard errors.
- `call`: a copy of the function call.
- `opt`: a list of saem algorithm control parameters.
- `conv_ind_cll`: vector of convergence indicator.
- `data_list`: a list of individual data used for estimation.

## Functions

- `print(rpinpall)`: Displays the distribution of estimated crop input uses accounting for error by default
- `summary(rpinpall)`: Displays a summary of estimated parameters
- `plot(rpinpall)`: Plot the "global" convergence indicator

## References

Koutchade, O. P., Carpentier A. and Femenia F. (2024).

## Examples

```
data(my_winputall_data)
mydata <- my_winputall_data
fit <- rpinpallEst(data = my_winputall_data,
                   id_time = c("id","year"),
                   total_input = "tx",
                   crop_acreage = c("s_crop1","s_crop2","s_crop3"),
                   distrib_method = "lognormal",
                   sim_method = "map_imh",
                   calib_method = "cmode",
                   saem_control = list(nb_SA = 10, nb_smooth = 10, estim_rdraw = 10))
print(fit)
plot(fit)
summary(fit)
head(fit$xit_pred)
```

---

`rp_model_fun` *Random Parameters Transformation*

---

## Description

rp.model.fun is used to tranform the random parameters.

## Usage

```
rp_model_fun(beta, opt)
```

## Arguments

| | |
|---|---|
| beta | vector of random parameters. beta follows multivariate normal distribution. |
| opt | a list of control parameters. See rpinpall. |

## Value

the transformed beta

# Index