

Package ‘agriPAM’

September 10, 2026

Type Package

Title Agricultural Policy Analysis Matrix Toolkit

Version 0.1.0

Description Builds and analyses Policy Analysis Matrices ('PAMs') for agricultural production systems. Computes private and social profitability, policy transfers, the domestic resource cost ratio, nominal protection coefficients for outputs and inputs, the effective protection coefficient, the private cost ratio, the profitability coefficient, subsidy ratios, and social cost-benefit ratios. Supports itemised farm budgets, parity prices, grouped analysis, deterministic sensitivity analysis, switching values, and correlated Monte Carlo simulation. Methods follow Monke and Pearson (1989, ISBN:0801419530) and the Food and Agriculture Organization of the United Nations (2007, ISBN:9789251057476).

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

Depends R (>= 4.1.0)

Imports graphics, grDevices, stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Classification/JEL Q12, Q18

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Chiranjit Mazumder [aut, cre],
Himadri Sekhar Roy [aut],
Utkarsh Tiwari [aut],
Pramit Pandit [aut],
Bikramjeet Ghose [aut]

Maintainer Chiranjit Mazumder <majumder.chira@gmail.com>

Repository CRAN
Date/Publication 2026-09-10 09:30:09 UTC

Contents

agriPAM-package	2
agri_pam_example	3
is_agri_pam	3
pam	4
pam_aggregate	5
pam_from_budget	6
pam_monte_carlo	7
pam_sensitivity	9
pam_values	10
parity_price	11
print.agri_pam	12
switching_value	13
Index	15

agriPAM-package	<i>agriPAM: Agricultural Policy Analysis Matrix Toolkit</i>
-----------------	---

Description

Build, inspect, and stress-test Policy Analysis Matrices (PAMs) for agricultural production systems. The package implements private, social, and transfer accounts and standard competitiveness, comparative-advantage, and policy-protection indicators.

Details

Create a matrix with `pam()` or `pam_from_budget()`, inspect it with `pam_matrix()`, `pam_transfers()`, and `pam_indicators()`, and then examine uncertainty with `pam_sensitivity()`, `switching_value()`, or `pam_monte_carlo()`.

References

Monke, E. A. and Pearson, S. R. (1989). *The Policy Analysis Matrix for Agricultural Development*. Cornell University Press. ISBN 0801419530.

Food and Agriculture Organization of the United Nations (2007). *Agricultural Trade Policy and Food Security in the Caribbean*. ISBN 9789251057476.

See Also

[pam](#), [pam_from_budget](#), [pam_monte_carlo](#)

`agri_pam_example`*Illustrative Agricultural PAM Data*

Description

Return an itemised, synthetic crop budget or the corresponding Policy Analysis Matrix. Values are for examples and teaching, not policy inference.

Usage

```
agri_pam_example(type = c("pam", "budget"))
```

Arguments

`type` Either "pam" (the default) or "budget".

Value

An `agri_pam` object for `type = "pam"`; otherwise a data frame with 28 itemised budget rows.

Examples

```
x <- agri_pam_example()
pam_indicators(x)
head(agri_pam_example("budget"))
```

`is_agri_pam`*Test for an agriPAM Object*

Description

Test whether an object inherits from class `agri_pam`.

Usage

```
is_agri_pam(x)
```

Arguments

`x` An object.

Value

A single logical value.

Examples

```
is_agri_pam(agri_pam_example())
is_agri_pam(data.frame())
```

pam

Construct a Policy Analysis Matrix

Description

Construct one or more agricultural Policy Analysis Matrices from private and social revenues and costs, directly or from columns in a data frame.

Usage

```
pam(
  private_revenue,
  private_tradable_inputs,
  private_domestic_factors,
  social_revenue,
  social_tradable_inputs,
  social_domestic_factors,
  id = NULL,
  unit = NULL,
  currency = NULL
)

pam_from_data(
  data,
  private_revenue,
  private_tradable_inputs,
  private_domestic_factors,
  social_revenue,
  social_tradable_inputs,
  social_domestic_factors,
  id = NULL,
  unit = NULL,
  currency = NULL
)
```

Arguments

private_revenue, private_tradable_inputs, private_domestic_factors	For pam(), private (observed market-price) revenue, tradable-input cost, and domestic-factor cost. For pam_from_data(), character strings naming those columns in data.
social_revenue, social_tradable_inputs, social_domestic_factors	For pam(), social (efficiency-price) revenue, tradable-input cost, and domestic-factor cost. For pam_from_data(), character strings naming those columns in data.
id	For pam(), optional analysis identifiers. A scalar is recycled and identifiers must be unique. For pam_from_data(), NULL or the name of an identifier column.

unit	Optional character label for the activity unit, such as "ha" or "tonne".
currency	Optional character label for the currency, such as "INR" or "USD".
data	A non-empty data frame containing PAM component columns.

Details

Every component vector must be non-negative and finite. Scalars are recycled to the common length. Private and social profits are revenue minus tradable-input cost minus domestic-factor cost.

Value

An object of class `agri_pam`.

Examples

```
x <- pam(
  private_revenue = 150000,
  private_tradable_inputs = 42000,
  private_domestic_factors = 61000,
  social_revenue = 140000,
  social_tradable_inputs = 46000,
  social_domestic_factors = 55000,
  id = "Paddy", unit = "ha", currency = "INR"
)
x
pam_matrix(x)

dat <- data.frame(
  crop = c("Paddy", "Wheat"),
  pr = c(150000, 110000), pt = c(42000, 31000), pd = c(61000, 44000),
  sr = c(140000, 116000), st = c(46000, 34000), sd = c(55000, 46000)
)
pam_from_data(dat, "pr", "pt", "pd", "sr", "st", "sd", id = "crop")
```

pam_aggregate

Aggregate Policy Analysis Matrices

Description

Sum the six additive PAM accounts within groups and recalculate profits, transfers, and ratios from the totals.

Usage

```
pam_aggregate(x, group = NULL, id = "All")
```

Arguments

x	An agri_pam object.
group	NULL for one overall matrix, or a grouping vector with one value per analysis in x.
id	Label used when group is NULL.

Value

An agri_pam object with one row per group.

Examples

```
x <- agri_pam_example()
pam_aggregate(x)
pam_aggregate(x, c("Cereal", "Cereal", "Pulse", "Oilseed"))
```

pam_from_budget	<i>Build a PAM from an Itemised Agricultural Budget</i>
-----------------	---

Description

Convert item quantities and private/social prices into revenue, tradable-input, and domestic-factor accounts, optionally for several production systems.

Usage

```
pam_from_budget(
  data,
  quantity,
  private_price,
  social_price,
  category,
  id = NULL,
  unit = NULL,
  currency = NULL
)
```

Arguments

data	A non-empty data frame containing budget items.
quantity, private_price, social_price, category	Character strings naming the quantity, private-price, social-price, and category columns.
id	NULL for one budget or a character string naming the production-system identifier column.
unit, currency	Optional labels passed to pam().

Details

Categories are normalised without regard to case, spaces, or hyphens. Accepted categories are output, tradable_input, and domestic_factor; documented aliases are also accepted. Values are quantity multiplied by price and summed within production system and category.

Value

An agri_pam object.

Examples

```
budget <- agri_pam_example("budget")
x <- pam_from_budget(
  budget, "quantity", "private_price", "social_price", "category",
  id = "crop", unit = "ha", currency = "INR"
)
pam_matrix(x, "Paddy")
```

pam_monte_carlo

Monte Carlo Uncertainty Analysis for a PAM

Description

Simulate uncertainty in the six additive PAM components using correlated lognormal or non-negative normal draws, then summarise policy indicators and decision probabilities.

Usage

```
pam_monte_carlo(
  x,
  cv = 0.1,
  n = 10000L,
  distribution = c("lognormal", "normal"),
  correlation = NULL,
  seed = NULL,
  index = 1L,
  probs = c(0.025, 0.5, 0.975)
)

## S3 method for class 'agri_pam_mc'
print(x, ...)

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

## S3 method for class 'agri_pam_mc'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

```
## S3 method for class 'agri_pam_mc'
plot(x, metric = "drc", breaks = "FD", ...)
```

Arguments

<code>x</code>	An <code>agri_pam</code> object, or an <code>agri_pam_mc</code> object for the methods.
<code>cv</code>	Coefficients of variation: one value, six unnamed values in standard component order, or a named subset. Named omitted components are fixed.
<code>n</code>	Number of Monte Carlo draws.
<code>distribution</code>	Either "lognormal" or "normal". Normal draws are truncated at zero.
<code>correlation</code>	Optional six-by-six correlation matrix for latent normal shocks.
<code>seed</code>	Optional integer seed. The caller's random-number state is restored on exit when a seed is supplied.
<code>index</code>	One numeric row number or character analysis identifier.
<code>probs</code>	Quantile probabilities used in the summary.
<code>object</code>	An <code>agri_pam_mc</code> object.
<code>row.names, optional</code>	Arguments for the data-frame method.
<code>metric</code>	Metric whose simulated distribution is plotted.
<code>breaks</code>	Histogram breaks passed to <code>graphics::hist()</code> .
<code>...</code>	Additional arguments passed to the underlying method.

Value

`pam_monte_carlo()` returns an `agri_pam_mc` object containing draws, summary, and probabilities data frames. `summary()` returns the latter two tables, `as.data.frame()` returns the draws, and the other methods return `x` invisibly.

Examples

```
x <- agri_pam_example()
mc <- pam_monte_carlo(
  x, cv = 0.10, n = 500, seed = 123, index = "Paddy"
)
mc
summary(mc)
plot(mc, metric = "drc")

cvs <- c(social_revenue = 0.15, social_tradable_inputs = 0.08)
pam_monte_carlo(x, cv = cvs, n = 250, seed = 42, index = "Wheat")
```

pam_sensitivity

Deterministic Sensitivity Analysis for a PAM

Description

Vary one PAM component by specified proportional changes and recalculate all profits, transfers, and policy indicators.

Usage

```
pam_sensitivity(
  x,
  parameter,
  changes = seq(-0.2, 0.2, by = 0.05),
  index = 1L
)

## S3 method for class 'agri_pam_sensitivity'
print(x, ...)

## S3 method for class 'agri_pam_sensitivity'
plot(x, metric = "drc", ...)
```

Arguments

x	An agri_pam object, or an agri_pam_sensitivity object for the methods.
parameter	One of the six private or social revenue, tradable-input, or domestic-factor component names.
changes	Numeric proportional changes; -0.1 is a 10 percent reduction and 0.2 a 20 percent increase.
index	One numeric row number or character analysis identifier.
metric	Metric to plot.
...	Additional arguments passed to the underlying method.

Value

pam_sensitivity() returns an object of class agri_pam_sensitivity containing a results data frame. Methods return x invisibly.

Examples

```
x <- agri_pam_example()
s <- pam_sensitivity(
  x, "social_revenue", changes = seq(-0.2, 0.2, by = 0.1),
  index = "Paddy"
)
```

```
s
plot(s, metric = "drc")
```

pam_values

Extract Policy Analysis Matrix Results

Description

Expose value accounts, policy transfers, standard indicators, qualitative classifications, or the conventional three-by-four PAM table.

Usage

```
pam_values(x)

pam_transfers(x)

pam_indicators(x)

pam_classify(x)

pam_matrix(x, index = 1L)
```

Arguments

x	An agri_pam object.
index	Numeric row numbers or character analysis identifiers.

Details

For the conventional matrix, private values are denoted by A–D, social values by E–H, and transfers by I–L:

$$D = A - B - C, \quad H = E - F - G, \quad L = D - H = I - J - K.$$

The indicator columns are:

- npco: nominal protection coefficient on output, A/E;
- npc_i: nominal protection coefficient on tradable inputs, B/F;
- epc: effective protection coefficient, (A-B)/(E-F);
- pcr: private cost ratio, C/(A-B);
- drc: domestic resource cost ratio, G/(E-F);
- pc: profitability coefficient, D/H;
- srp: subsidy ratio to producers, L/E;
- scb: social cost-benefit ratio, (F+G)/E.

A ratio is NA when its denominator is zero.

Value

pam_values(), pam_transfers(), pam_indicators(), and pam_classify() return data frames. pam_matrix() returns one numeric matrix, or a named list of matrices if multiple analyses are selected.

Examples

```
x <- agri_pam_example()
pam_values(x)
pam_transfers(x)
pam_indicators(x)
pam_classify(x)
pam_matrix(x, "Paddy")
```

parity_price

Calculate an Import- or Export-Parity Price

Description

Calculate import- or export-parity prices for social valuation.

Usage

```
parity_price(
  border_price,
  exchange_rate = 1,
  transport = 0,
  handling = 0,
  other_costs = 0,
  quality_adjustment = 1,
  direction = c("import", "export")
)
```

Arguments

border_price Border price in international currency per unit.
exchange_rate Domestic currency units per international currency unit.
transport, handling, other_costs
 Domestic marketing costs per unit.
quality_adjustment
 Multiplicative adjustment for quality differences.
direction Either "import" or "export" parity.

Details

Import parity adds domestic transfer costs to the quality-adjusted border value. Export parity subtracts them. Taxes and subsidies should not be included when constructing an undistorted social price.

Value

A numeric vector of parity prices in domestic currency per unit.

Examples

```
parity_price(300, exchange_rate = 83, transport = 1200,
             handling = 300, direction = "import")
parity_price(300, exchange_rate = 83, transport = 1200,
             handling = 300, direction = "export")
```

print.agri_pam	<i>Methods for agriPAM Objects</i>
----------------	------------------------------------

Description

Print, summarise, plot, or coerce Policy Analysis Matrices.

Usage

```
## S3 method for class 'agri_pam'
print(x, ...)

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

## S3 method for class 'summary_agri_pam'
print(x, ...)

## S3 method for class 'agri_pam'
plot(
  x,
  type = c("profits", "coefficients", "transfers"),
  index = NULL,
  main = NULL,
  ...
)

## S3 method for class 'agri_pam'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

- x, object An agri_pam object, or a summary_agri_pam object for its print method.
- type Plot type: profits, coefficients, or transfers.
- index Optional numeric row numbers or character analysis identifiers.
- main Optional plot title.

```

row.names, optional
                Arguments for the data-frame method.
...
                Additional arguments passed to the underlying method.

```

Value

Print and plot methods return `x` invisibly. `summary()` returns a `summary_agri_pam` object containing an overview data frame. `as.data.frame()` returns a combined results data frame.

Examples

```

x <- agri_pam_example()
print(x)
summary(x)
as.data.frame(x)
plot(x, type = "coefficients")

```

switching_value	<i>Find a PAM Switching Value</i>
-----------------	-----------------------------------

Description

Find the nearest proportional change in one PAM component that makes a selected profit, transfer, or ratio reach a target.

Usage

```

switching_value(
  x,
  parameter,
  metric = "drc",
  target = NULL,
  interval = c(-0.95, 3),
  index = 1L,
  tol = 1e-08
)

## S3 method for class 'agri_pam_switch'
print(x, ...)

```

Arguments

<code>x</code>	An <code>agri_pam</code> object, or an <code>agri_pam_switch</code> object for the print method.
<code>parameter</code>	One of the six private or social revenue, tradable-input, or domestic-factor component names.
<code>metric</code>	A profit, transfer, or indicator name.

target	Numeric target. If NULL, zero is used for profits, transfers, and srp, and one for other ratios.
interval	Two proportional changes defining the search interval.
index	One numeric row number or character analysis identifier.
tol	Numeric root-finding tolerance.
...	Additional arguments passed to the print method.

Value

A one-row data frame of class `agri_pam_switch`. The found column is FALSE when the target is not reached inside interval.

Examples

```
x <- agri_pam_example()
switching_value(x, "social_revenue", metric = "drc", index = "Paddy")
switching_value(
  x, "private_revenue", metric = "private_profit", index = "Paddy"
)
```

Index

- * **package**
 - agriPAM-package, 2
- agri_pam_example, 3
- agriPAM (agriPAM-package), 2
- agriPAM-package, 2
- as.data.frame.agri_pam
 - (print.agri_pam), 12
- as.data.frame.agri_pam_mc
 - (pam_monte_carlo), 7
- is_agri_pam, 3
- pam, 2, 4
- pam_aggregate, 5
- pam_classify (pam_values), 10
- pam_from_budget, 2, 6
- pam_from_data (pam), 4
- pam_indicators (pam_values), 10
- pam_matrix (pam_values), 10
- pam_monte_carlo, 2, 7
- pam_sensitivity, 9
- pam_transfers (pam_values), 10
- pam_values, 10
- parity_price, 11
- plot.agri_pam (print.agri_pam), 12
- plot.agri_pam_mc (pam_monte_carlo), 7
- plot.agri_pam_sensitivity
 - (pam_sensitivity), 9
- print.agri_pam, 12
- print.agri_pam_mc (pam_monte_carlo), 7
- print.agri_pam_sensitivity
 - (pam_sensitivity), 9
- print.agri_pam_switch
 - (switching_value), 13
- print.summary_agri_pam
 - (print.agri_pam), 12
- summary.agri_pam (print.agri_pam), 12
- summary.agri_pam_mc (pam_monte_carlo), 7
- switching_value, 13