Package 'farmPartial'

September 9, 2026

Type Package

Title Tools for Farm Partial-Budget Analysis

Version 0.1.0

Description Provides tools for evaluating the incremental economic consequences of a proposed farm-management change using partial-budget logic. Functions organize added returns, reduced costs, added costs, and reduced returns; compare baseline and alternative budgets; calculate net changes and marginal rates of return; conduct one- and two-way sensitivity, scenario, break-even, dominance, marginal, and Monte Carlo uncertainty analyses; and convert capital changes to annual equivalents. The framework follows the approach described by the International Maize and Wheat Improvement Center (1988, ISBN: 968-6127-19-4) for farm-management, extension, and on-farm research.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports graphics, stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Chiranjit Mazumder [aut, cre, cph],
Utkarsh Tiwari [aut, cph],
Anbukkani Perumal [aut, cph]

Maintainer Chiranjit Mazumder <chiranjit@iari.res.in>

Repository CRAN

Date/Publication 2026-09-09 14:40:10 UTC

Contents

`farmPartial-package` *farmPartial: Tools for Farm Partial-Budget Analysis*

Description

Tools for estimating the incremental economic consequence of a farm-management change. The package implements the four standard partial-budget components: added returns, reduced costs, added costs, and reduced returns. It also supports baseline-alternative comparisons, sensitivity and scenario analysis, break-even analysis, Monte Carlo uncertainty analysis, capital annualization, dominance analysis, and marginal rate-of-return analysis.

Details

The central identity is

$$\Delta NR = (AR + RC) - (AC + RR),$$

where AR is added returns, RC is reduced costs, AC is added costs, and RR is reduced returns. Only items that change between the baseline and alternative farm plans belong in a partial budget.

References

CIMMYT (1988). *From Agronomic Data to Farmer Recommendations: An Economics Training Manual*. Completely revised edition. Mexico: CIMMYT. ISBN 968-6127-19-4.

See Also

`partial_budget`, `compare_budgets`, `sensitivity_analysis`, `simulate_partial_budget`

annualize_investment *Annualize a Farm Capital Investment*

Description

Converts a farm capital purchase and terminal salvage value into an equivalent annual cost using the capital-recovery method.

Usage

```
annualize_investment(purchase, salvage = 0, life, rate = 0)
```

Arguments

purchase	Non-negative purchase cost.
salvage	Non-negative expected salvage value, not greater than purchase cost.
life	Positive useful life in years.
rate	Non-negative annual discount rate expressed as a decimal.

Details

For a positive discount rate, the function first discounts the terminal salvage value to the present and then applies the capital-recovery factor. At a zero discount rate it returns straight-line annual capital consumption, $(P - S)/n$.

Value

Numeric vector of equivalent annual costs.

Examples

```
annualize_investment(purchase = 120000, salvage = 20000, life = 8, rate = 0.08)
```

break_even_component *Break-Even Value for One Partial-Budget Component*

Description

Calculates the selected partial-budget component amount required to make net economic change equal zero.

Usage

```
break_even_component(x, item)
```

Arguments

x	A farm_partial_budget object.
item	Exact label of a uniquely occurring component with a non-zero current amount.

Details

All other budget components are held fixed. If the calculated break-even amount is negative, the returned feasible_nonnegative flag is FALSE; this means no non-negative amount for the selected component can make the current budget exactly break even while all other items remain unchanged.

Value

A one-row data frame containing the break-even amount and multiplier.

Examples

```
pb <- partial_budget(wheat_example("changes"))
break_even_component(pb, "Additional herbicide")
```

---

compare_budgets *Compare Baseline and Alternative Farm Budgets*

Description

Compares baseline and alternative farm budgets and classifies their incremental economic changes.

Usage

```
compare_budgets(
  baseline,
  alternative,
  currency = NULL,
  unit = NULL,
  zero_tol = sqrt(.Machine$double.eps)
)
```

Arguments

baseline, alternative	Data frames containing item, category, and value; objects from farm_budget() are recommended.
currency, unit	Optional labels overriding those stored on farm-budget objects.
zero_tol	Non-negative numeric tolerance below which a difference is treated as zero.

Details

For return items, an increase is classified as an added return and a decrease as a reduced return. For cost items, an increase is an added cost and a decrease is a reduced cost. An item absent from one budget is assigned value zero there. An item must retain the same return/cost category across both plans.

Value

A `farm_partial_budget` object. Its `comparison` element contains the aligned baseline and alternative values and their signed differences.

Examples

```
base <- wheat_example("baseline")
alt <- wheat_example("alternative")
compare_budgets(base, alt)
```

`farm_budget` *Build a Farm Budget Table*

Description

Creates a validated farm budget from total values or quantity-price components.

Usage

```
farm_budget(
  item,
  category,
  value = NULL,
  quantity = NULL,
  unit_price = NULL,
  currency = "INR",
  unit = "per ha"
)

## S3 method for class 'farm_budget'
print(x, ..., row.names = FALSE)
```

Arguments

`item`	Character vector of budget-item labels.
`category`	Character vector identifying each item as a return/revenue or a cost/expense.
`value`	Optional non-negative total value for each item.
`quantity`, `unit_price`	Optional non-negative quantities and unit prices. When `value` is omitted, both must be supplied.

currency	Currency label.
unit	Scale label.
x	A farm_budget object.
...	Additional arguments passed to print.data.frame.
row.names	Logical controlling printing of row names.

Value

A data frame of class farm_budget.

Examples

```
b <- farm_budget(
  item = c("Grain", "Seed"),
  category = c("return", "cost"),
  value = c(100000, 7000)
)
b
```

---

partial_budget *Create and Summarize a Farm Partial Budget*

Description

Organizes the four partial-budget components and calculates the expected change in net return.

Usage

```
partial_budget(
  changes = NULL,
  added_returns = NULL,
  reduced_costs = NULL,
  added_costs = NULL,
  reduced_returns = NULL,
  currency = "INR",
  unit = "per ha"
)

budget_summary(x)

## S3 method for class 'farm_partial_budget'
print(x, digits = 2L, ...)

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

## S3 method for class 'farm_partial_budget'
```

```
plot(
  x,
  main = "Farm partial budget",
  ylab = NULL,
  col = c("#2E7D32", "#66BB6A", "#C62828", "#EF5350", "#1565C0"),
  ...
)
```

Arguments

changes	Optional data frame with columns item, change_type, and amount.
added_returns, reduced_costs, added_costs, reduced_returns	Optional named non-negative numeric vectors. These provide a compact alternative to changes.
currency	Single character currency label.
unit	Single character scale label, such as "per ha".
x, object	A farm_partial_budget object.
digits	Number of decimal places printed.
main	Plot title.
ylab	Plot y-axis label; generated from currency and unit if omitted.
col	Bar colors.
...	Additional arguments passed to the relevant print or plotting method.

Details

The four accepted change types are added_return, reduced_cost, added_cost, and reduced_return. Amounts are magnitudes and must be non-negative. The estimated net change is favorable changes minus adverse changes. The change benefit-cost ratio is the total favorable changes divided by total adverse changes. The marginal rate of return is reported only when the net change in costs is positive and is calculated as net change divided by the increase in costs, multiplied by 100.

Value

partial_budget() returns a farm_partial_budget object. budget_summary() and summary() return a one-row data frame. The print and plot methods return their input invisibly or the plotting positions invisibly.

Examples

```
pb <- partial_budget(
  added_returns = c("Extra grain" = 6000),
  reduced_costs = c("Energy saving" = 1800),
  added_costs = c("New input" = 2200),
  reduced_returns = c("Lower straw value" = 500),
  currency = "INR",
  unit = "per ha"
)
pb
budget_summary(pb)
```

scenario_analysis *Scenario Analysis for a Farm Partial Budget*

Description

Applies item-specific multipliers under named scenarios and recalculates partial-budget outcomes.

Usage

```
scenario_analysis(x, scenarios, include_base = TRUE)

## S3 method for class 'farm_partial_scenario'
print(x, ..., row.names = FALSE)

## S3 method for class 'farm_partial_scenario'
plot(
  x,
  main = "Partial-budget scenarios",
  ylab = NULL,
  col = NULL,
  ...
)
```

Arguments

x	A farm_partial_budget object, or a scenario result for print and plot methods.
scenarios	Data frame with scenario, item, and multiplier columns.
include_base	Logical; include the unmodified partial budget as "Base".
main, ylab, col	Plot title, y-axis label, and colors.
...	Additional print or plot arguments.
row.names	Logical controlling printing of row names.

Value

A data frame of class farm_partial_scenario.

Examples

```
pb <- partial_budget(wheat_example("changes"))
sc <- data.frame(
  scenario = c("Low output price", "High input price"),
  item = c("Higher grain return", "Additional herbicide"),
  multiplier = c(0.8, 1.25)
)
scenario_analysis(pb, sc)
```

`sensitivity_analysis` *One-Way Sensitivity Analysis for a Partial Budget*

Description

Varies one partial-budget component while holding all remaining components constant.

Usage

```
sensitivity_analysis(x, item, multipliers = seq(0.8, 1.2, by = 0.05))

## S3 method for class 'farm_partial_sensitivity'
print(x, ..., row.names = FALSE)

## S3 method for class 'farm_partial_sensitivity'
plot(
  x,
  main = "One-way sensitivity analysis",
  xlab = "Multiplier",
  ylab = NULL,
  col = "#1565C0",
  ...
)
```

Arguments

`x`	A `farm_partial_budget` object, or a sensitivity result for the print and plot methods.
`item`	Exact label of a uniquely occurring budget item.
`multipliers`	Non-negative numeric multipliers applied to that item.
`main`, `xlab`, `ylab`, `col`	Plot labels and color.
`...`	Additional print or plot arguments.
`row.names`	Logical controlling printing of row names.

Value

A data frame of class `farm_partial_sensitivity`.

Examples

```
pb <- partial_budget(wheat_example("changes"))
s <- sensitivity_analysis(pb, "Higher grain return", c(0.8, 1, 1.2))
s
```

`simulate_partial_budget` *Monte Carlo Uncertainty Analysis for a Partial Budget*

Description

Propagates uncertainty in partial-budget components through Monte Carlo simulation.

Usage

```
simulate_partial_budget(x, uncertainty, n = 10000L, level = 0.95, seed = NULL)

## S3 method for class 'farm_partial_simulation'
print(x, digits = 2L, ...)

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

## S3 method for class 'farm_partial_simulation'
plot(
  x,
  main = "Uncertainty in partial-budget net change",
  xlab = NULL,
  col = "#90CAF9",
  border = "white",
  ...
)
```

Arguments

`x`, `object`	A `farm_partial_budget` object, or a simulation object for its methods.
`uncertainty`	Data frame with `item` and `distribution`, plus relevant parameter columns among `mean`, `sd`, `cv`, `min`, `mode`, and `max`.
`n`	Integer number of Monte Carlo draws, at least two.
`level`	Probability level strictly between zero and one for the central uncertainty interval.
`seed`	Optional finite numeric seed.
`digits`	Number of decimal places printed.
`main`, `xlab`, `col`, `border`	Histogram title, axis label, fill color, and border color.
`...`	Additional arguments passed to print or histogram methods.

Details

Supported distributions are `fixed`, `normal`, `lognormal`, `uniform`, and `triangular`. A normal or lognormal component uses `mean` (defaulting to its current budget amount) and either `sd` or `cv`. Uniform components require `min` and `max`; triangular components require `min`, `mode`, and `max`. Component draws are independent. Negative normal draws are truncated at zero. The returned interval describes the simulated uncertainty distribution; it is not a sampling-theory confidence interval.

Value

An object of class `farm_partial_simulation`. Its `draws` element contains simulated net changes; `component_draws` contains the simulated component amounts; and `metrics` contains mean, median, standard deviation, interval endpoints, and probability of a positive net change.

Examples

```
pb <- partial_budget(wheat_example("changes"))
u <- data.frame(
  item = c("Higher grain return", "Additional herbicide"),
  distribution = c("normal", "uniform"),
  mean = c(6000, NA),
  sd = c(900, NA),
  min = c(NA, 900),
  max = c(NA, 1500)
)
sim <- simulate_partial_budget(pb, u, n = 500, seed = 123)
summary(sim)
```

`trial_budget` *Economic Analysis of Alternative Farm Treatments*

Description

Calculates treatment economics and supports dominance and marginal rate-of-return analyses.

Usage

```
trial_budget(treatment, yield, price, variable_cost, yield_adjustment = 1)

dominance_analysis(x)

marginal_analysis(x, minimum_mrr = NULL)
```

Arguments

`treatment`	Unique non-empty treatment labels.
`yield`	Non-negative observed yield.

price	Non-negative farm-gate output price per yield unit.
variable_cost	Non-negative total cost that varies by treatment.
yield_adjustment	Positive factor no greater than one for adjusting experimental yield to expected farm conditions.
x	Data frame with at least treatment, variable_cost, and net_benefit columns, such as the result from trial_budget().
minimum_mrr	Optional non-negative minimum acceptable marginal rate of return, expressed as a percentage.

Details

dominance_analysis() marks an alternative as dominated when another has no greater variable cost and no lower net benefit, with at least one strict improvement. marginal_analysis() removes dominated alternatives, sorts the remainder by variable cost, and computes the change in net benefit divided by the change in variable cost between adjacent alternatives, multiplied by 100.

Value

trial_budget() returns a treatment-economics data frame. dominance_analysis() adds a logical dominated column. marginal_analysis() returns only non-dominated alternatives with marginal increments and marginal rates of return.

References

CIMMYT (1988). *From Agronomic Data to Farmer Recommendations: An Economics Training Manual*. Completely revised edition. Mexico: CIMMYT. ISBN 968-6127-19-4.

Examples

```
trials <- trial_budget(
  treatment = c("Farmer practice", "Treatment A", "Treatment B"),
  yield = c(3.0, 3.5, 3.7),
  price = 22000,
  variable_cost = c(18000, 23000, 31000),
  yield_adjustment = 0.9
)
dominance_analysis(trials)
marginal_analysis(trials, minimum_mrr = 50)
```

---

two_way_sensitivity *Two-Way Sensitivity Analysis for a Partial Budget*

Description

Varies two partial-budget components simultaneously and calculates the resulting net changes.

Usage

```
two_way_sensitivity(
  x,
  item_x,
  item_y,
  multipliers_x = seq(0.8, 1.2, by = 0.1),
  multipliers_y = seq(0.8, 1.2, by = 0.1)
)

## S3 method for class 'farm_partial_two_way'
print(x, ..., row.names = FALSE)

## S3 method for class 'farm_partial_two_way'
plot(
  x,
  main = "Two-way sensitivity: net change",
  xlab = NULL,
  ylab = NULL,
  col = c("#A50026", "#D73027", "#F46D43", "#FDAE61", "#FEE08B",
    "#D9EF8B", "#A6D96A", "#66BD63", "#1A9850", "#006837"),
  ...
)
```

Arguments

x	A farm_partial_budget object, or a two-way sensitivity result for the print and plot methods.
item_x, item_y	Exact labels of two different uniquely occurring budget items.
multipliers_x, multipliers_y	Non-negative multiplier vectors.
main, xlab, ylab, col	Plot labels and colors.
...	Additional print or plot arguments.
row.names	Logical controlling printing of row names.

Value

A data frame of class farm_partial_two_way.

Examples

```
pb <- partial_budget(wheat_example("changes"))
two_way_sensitivity(
  pb,
  "Higher grain return",
  "Additional herbicide",
  multipliers_x = c(0.9, 1, 1.1),
  multipliers_y = c(0.9, 1, 1.1)
```

```
)
```

`wheat_example` *Illustrative Wheat Partial-Budget Data*

Description

Returns internally generated per-hectare wheat-management values for package examples. The values are pedagogical and are not survey estimates, official costs, or recommendations.

Usage

```
wheat_example(type = c("changes", "baseline", "alternative"))
```

Arguments

`type`	One of `"changes"`, `"baseline"`, or `"alternative"`.

Value

For `"changes"`, a data frame accepted by `partial_budget()`. For the other choices, a `farm_budget` object accepted by `compare_budgets()`.

Examples

```
partial_budget(wheat_example("changes"))
compare_budgets(wheat_example("baseline"), wheat_example("alternative"))
```

Index