Package: econagent (via r-universe)

July 14, 2026

Type Package

Title Utility Functions and Composite Goods in Economics

Version 0.1.0

Description Provides utility functions and composite goods for building economic models.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

Suggests testthat (>= 3.0.0), tibble, tidyr

Config/testthat/edition 3

Imports cli, dplyr, FixedPoint, lifecycle, partialised, pillar, purrr, rlang, timbr (>= 0.2.2.9000), vctrs

Depends R (>= 2.10)

Config/pak/sysreqs libglpk-dev libicu-dev libxml2-dev

Repository https://uchidamizuki.r-universe.dev

Date/Publication 2026-01-15 14:45:36 UTC

RemoteUrl https://github.com/UchidaMizuki/econagent

RemoteRef HEAD

RemoteSha 3531c8d148fffa5dc31175aad1882286f8ae228d

Contents

`goods_by` *Create goods*

Description

Create goods

Usage

```
goods_by(.data, ...)
```

Arguments

`.data`	A data frame that contains columns `price` and `quantity`. If `price` is not available, it will be created with a value of 1.
`...`	Variables to group by.

Value

A `econ_goods` object.

goods_compose *Compose goods*

Description

Compose goods

Usage

```
goods_compose(data, utility, node = NULL)
```

Arguments

data	A econ_goods object.
utility	A econ_util object or a data frame that contains columns utility.
node	A node name for composition goods. By default, it is NULL.

Value

A econ_goods object.

goods_consume *Consume goods*

Description

Consume goods

Usage

```
goods_consume(data, incomes = NULL)
```

Arguments

data	A econ_goods object.
incomes	A data frame of a list of data frames that contains columns income.

Value

A econ_goods object.

goods_consume_recursively *Consume goods recursively*

Description

Consume goods recursively

Usage

```
goods_consume_recursively(data, f, ...)
```

Arguments

data	A econ_goods object.
f	A function that returns a data frame with columns income.
...	Additional arguments passed to stats::optim().

Value

A econ_goods object.

goods_produce *Produce goods*

Description

Produce goods

Usage

```
goods_produce(data, quantities = NULL)
```

Arguments

data	A econ_goods object.
quantities	A data frame of a list of data frames that contains columns quantity.

Value

A econ_goods object.

goods_produce_recursively *Produce goods recursively*

Description

Produce goods recursively

Usage

```
goods_produce_recursively(data, f, ...)
```

Arguments

data	A econ_goods object.
f	A function that returns a data frame with columns quantity.
...	Additional arguments passed to stats::optim().

Value

A econ_goods object.

goods_reprice *Reprice goods*

Description

Reprice goods

Usage

```
goods_reprice(data, prices = NULL)
```

Arguments

data	A econ_goods object.
prices	A data frame or a list of data frames that contains columns price.

Value

A econ_goods object.

goods_reprice_recursively *Reprice goods recursively*

Description

Reprice goods recursively

Usage

```
goods_reprice_recursively(data, f, ...)
```

Arguments

data	A econ_goods object.
f	A function that returns a data frame with columns price.
...	Additional arguments passed to stats::optim().

Value

A econ_goods object.

goods_trade_iceberg *Iceberg trade cost*

Description

Iceberg trade cost

Usage

```
goods_trade_iceberg(data, trade)
```

Arguments

data	A econ_goods object.
trade	A data frame with cost column.

Value

A econ_goods object.

goods_trade_update *Update trade cost*

Description

Update trade cost

Usage

```
goods_trade_update(data, trade)
```

Arguments

data	A econ_goods object.
trade	A data frame with cost column.

Value

A econ_goods object.

goods_update *Update goods data*

Description

Update goods data

Usage

```
goods_update(data, value)
```

Arguments

data	A econ_goods object.
value	A data frame with new values.

Value

A econ_goods object.

new_util *Create a new utility function*

Description

Create a new utility function

Usage

```
new_util(f, ..., class = character())
```

Arguments

f	A function that returns the utility.
...	Parameters to be passed to f.
class	Name of subclass.

Value

A econ_util object.

new_util_homothetic *Create a new homothetic utility function*

Description

Create a new homothetic utility function

Usage

```
new_util_homothetic(f, ..., class = character())
```

Arguments

f	A function that returns the utility.
...	Parameters to be passed to f.
class	Name of subclass.

Value

A util_homothetic object.

util_2goods_budget *Budget line function factory for two goods*

Description

[Experimental]

Usage

```
util_2goods_budget(prices, income)
```

Arguments

prices	A numeric vector of length 2 with the prices of the goods.
income	A scalar numeric of income.

Value

A function that takes a scalar numeric of quantity of good X and returns a scalar numeric of quantity of good Y.

util_2goods_indifference *Indifference curve function factory for two goods*

Description

[Experimental]

Usage

```
util_2goods_indifference(
  f,
  utility,
  otherwise = NA_real_,
  interval = c(1e-06, 1e+06),
  tol = 1e-06,
  ...
)
```

Arguments

f	A econ_util object.
utility	A scalar numeric of utility.
otherwise	Default value when the root is not found. By default, NA_real.
interval	Passed to stats::uniroot().
tol	Passed to stats::uniroot().
...	Passed to stats::uniroot().

Value

A function that takes a scalar numeric of quantity of good X and returns a scalar numeric of quantity of good Y.

---

util_2goods_utility *Utility function factory for two goods with a given quantity of good Y*

Description

[Experimental]

Usage

```
util_2goods_utility(f, quantity, gradient = FALSE)
```

Arguments

f	A econ_util object.
quantity	A scalar numeric of quantity.
gradient	Logical input to return the gradient. By default, FALSE.

Value

A function that takes a scalar numeric of quantity of good X and returns a scalar numeric of total utility (gradient = TRUE) or marginal utility (gradient = FALSE).

util_calibrate *Calibrate a utility function*

Description

Calibrate a utility function

Usage

```
util_calibrate(f, prices, quantities, ...)
```

Arguments

f	A econ_util object.
prices	A numeric vector of prices.
quantities	A numeric vector of quantities.
...	Additional arguments.

Value

A econ_util object with calibrated parameters.

util_ces *Constant elasticity of substitution (CES) utility function*

Description

Constant elasticity of substitution (CES) utility function

Usage

```
util_ces(
  substitution,
  homogeneity = 1,
  efficiency = NA_real_,
  weights = double(),
  homothetic = homogeneity == 1
)

util_ces2(
  elasticity_of_substitution,
  homogeneity = 1,
  efficiency = NA_real_,
  weights = double(),
  homothetic = homogeneity == 1
)
```

Arguments

substitution	A scalar numeric of substitution parameter.
homogeneity	A scalar numeric of degree of homogeneity. By default, 1. When homothetic = TRUE, homogeneity must be equal to 1.
efficiency	A scalar numeric of efficiency parameter.
weights	A numeric vector of weights.
homothetic	A logical scalar. By default, homogeneity == 1.
elasticity_of_substitution	A scalar numeric of elasticity of substitution.

Value

A util_ces object.

---

util_cobb_douglas *Cobb-Douglas utility function*

Description

Cobb-Douglas utility function

Usage

```
util_cobb_douglas(
  efficiency = NA_real_,
  weights = double(),
  homothetic = length(weights) == 0 || sum(weights) == 1
)
```

Arguments

efficiency	A scalar numeric of efficiency parameter. By default, NA_real_.
weights	A numeric vector of weight parameters. By default, double().
homothetic	A logical scalar. By default, TRUE.

Value

A util_cobb_douglas object.

util_demand *Demand function*

Description

This function works as Marshallian demand function when `income` is input, and as Hicksian demand function when `utility` is input.

Usage

```
util_demand(f, prices, income = NULL, utility = NULL, gradient = FALSE, ...)
```

Arguments

`f`	A `econ_util` object.
`prices`	A numeric vector of prices.
`income`	A scalar numeric of income. If `NULL`, `utility` must be provided.
`utility`	A scalar numeric of utility level. If `NULL`, `income` must be provided.
`gradient`	Logical input to return the gradient. By default, `FALSE`.
`...`	Additional arguments.

Value

when `gradient = FALSE`, a numeric vector of quantities. When `gradient = TRUE`, a numeric matrix of gradients of quantities related to prices.

util_demand_hicksian *Hicksian demand function*

Description

Hicksian demand function

Usage

```
util_demand_hicksian(f, prices, utility, gradient = FALSE, ...)
```

Arguments

`f`	A `econ_util` object.
`prices`	A numeric vector of prices.
`utility`	A scalar numeric of utility level.
`gradient`	Logical input to return the gradient. By default, `FALSE`.
`...`	Additional arguments.

Value

when `gradient = FALSE`, a numeric vector of quantities. When `gradient = TRUE`, a numeric matrix of gradients of quantities related to prices.

---

util_demand_marshallian *Marshallian demand function*

Description

Marshallian demand function

Usage

```
util_demand_marshallian(f, prices, income, gradient = FALSE, ...)
```

Arguments

`f`	A `econ_util` object.
`prices`	A numeric vector of prices.
`income`	A scalar numeric of income.
`gradient`	Logical input to return the gradient. By default, `FALSE`.
`...`	Additional arguments.

Value

when `gradient = FALSE`, a numeric vector of quantities. When `gradient = TRUE`, a numeric matrix of gradients of quantities related to prices.

---

util_expenditure *Expenditure function*

Description

Expenditure function

Usage

```
util_expenditure(f, prices, utility, gradient = FALSE, ...)
```

Arguments

f	A econ_util object.
prices	A numeric vector of prices.
utility	A scalar numeric of utility level.
gradient	Logical input to return the gradient. By default, FALSE.
...	Additional arguments.

Value

When gradient = FALSE, a scalar numeric of expenditure. When gradient = TRUE, a numeric vector of gradients of expenditure related to prices.

---

util_gradient *Gradient of a utility function*

Description

Gradient of a utility function

Usage

```
util_gradient(f, quantities, ...)
```

Arguments

f	A econ_util object.
quantities	A numeric vector of quantities.
...	Additional arguments.

Value

A numeric vector of gradients.

util_indirect *Indirect utility function*

Description

Indirect utility function

Usage

```
util_indirect(f, prices, income, gradient = FALSE, ...)
```

Arguments

f	A econ_util object.
prices	A numeric vector of prices.
income	A scalar numeric of income.
gradient	Logical input to return the gradient. By default, FALSE.
...	Additional arguments.

Value

When gradient = FALSE, a scalar numeric of utility level. When gradient = TRUE, a numeric vector of gradients of utility level related to prices.

util_leontief *Leontief utility function*

Description

Leontief utility function

Usage

```
util_leontief(efficiency = NA_real_, weights = double())
```

Arguments

efficiency	A scalar numeric of efficiency parameter. By default, NA_real_.
weights	A numeric vector of weight parameters. By default, double().

Value

A util_leontief object.

util_linear *Linear utility function*

Description

Linear utility function

Usage

```
util_linear(efficiency = NA_real_, weights = double())
```

Arguments

efficiency	A scalar numeric of efficiency parameter. By default, NA_real_.
weights	A numeric vector of weight parameters. By default, double().

Value

A util_linear object.

util_price *Price function*

Description

[Experimental]

Usage

```
util_price(f, prices, quantities, ...)
```

Arguments

f	A econ_util object.
prices	A numeric vector of prices.
quantities	A numeric vector of quantities.
...	Additional arguments.

Value

A numeric scalar of price.

Index