Package: wefnexus (via r-universe)

June 1, 2026

Type Package

Title Water-Energy-Food-Nutrient-Carbon Nexus Analysis for Agronomic Systems

Version 1.0.0

Description Provides functions for analysing Water-Energy-Food-Nutrient-Carbon (WEFNC) nexus interactions in agricultural production systems. Includes functions for computing water use efficiency (WUE), water productivity (WP), and water footprint (WF) including green, blue, and grey components following the methodology of Hoekstra et al. (2011, ISBN:9781849712798). Includes energy budgeting tools for energy use efficiency (EUE), energy return on investment (EROI), net energy (NE), and energy productivity (EP). Computes nutrient use efficiency (NUE) metrics including agronomic efficiency (AE), physiological efficiency (PE), recovery efficiency (RE), and partial factor productivity (PFP) as defined by Dobermann (2007) <https://digitalcommons.unl.edu/agronomyfacpub/316/> and Congreves et al. (2021) <doi:10.3389/fpls.2021.637108>. Estimates carbon footprint (CF), greenhouse gas (GHG) emissions, soil organic carbon (SOC) stocks, and global warming potential (GWP) using Intergovernmental Panel on Climate Change (IPCC) Sixth Assessment Report (AR6) default values (CH4 = 27, N2O = 273) as reported in Forster et al. (2021) <doi:10.1017/9781009157896.009>. Computes composite Water-Energy-Food-Nutrient-Carbon (WEFNC) nexus indices, trade-off correlation matrices, and generates radar and heatmap visualizations for comparing agricultural treatments. Supports conservation agriculture (CA), irrigated and rain-fed systems, and arid and semi-arid production environments. Methods follow Lal (2004) <doi:10.1016/j.envint.2004.03.005> for carbon emissions from farm operations, and Hoover et al. (2023) <doi:10.1016/j.scitotenv.2022.160992> for water use efficiency indicators.

License GPL (>= 3)

URL https://github.com/lalitrolaniya/wefnexus

BugReports https://github.com/lalitrolaniya/wefnexus/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports stats, graphics, grDevices, rlang

Suggests ggplot2, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Lalit Kumar Rolaniya [aut, cre] (ORCID: <https://orcid.org/0000-0001-8908-1211>), Hemant Poonia [aut] (ORCID: <https://orcid.org/0000-0002-0945-8793>), Ram Lal Jat [aut] (ORCID: <https://orcid.org/0009-0003-4339-0555>), Monika Punia [aut] (ORCID: <https://orcid.org/0009-0002-0294-6767>), Raja Ram Choudhary [aut]

Maintainer Lalit Kumar Rolaniya <lalitrolaniya@gmail.com>

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

Date/Publication 2026-04-02 13:32:34 UTC

RemoteUrl https://github.com/lalitrolaniya/wefnexus

RemoteRef HEAD

RemoteSha 8a19bdc73dd5e81aa34347ef59be3fe36316bad3

Contents

wefnexus-package	*wefnexus: Water-Energy-Food-Nutrient-Carbon Nexus Analysis for Agronomic Systems*

Description

Provides functions for analysing Water-Energy-Food-Nutrient-Carbon (WEFNC) nexus interactions in agricultural systems. The package provides five core modules and a nexus integration layer:

Water Water use efficiency (WUE), water productivity (WP), water footprint (WF) with green/blue/grey components, irrigation efficiency, crop water stress index (CWSI), and crop water productivity (CWP).

Energy Energy budgeting including input-output analysis, energy use efficiency (EUE), energy return on investment (EROI), energy productivity (EP), net energy (NE), specific energy, and energy profitability.

Food Food productivity indices, harvest index (HI), land equivalent ratio (LER), system productivity index (SPI), caloric and protein yield, and production efficiency index (PEI).

Nutrient Nutrient use efficiency (NUE) metrics including agronomic efficiency (AE), physiological efficiency (PE), recovery efficiency (RE), partial factor productivity (PFP), internal utilization efficiency (IUE), and nutrient harvest index (NHI).

Carbon Carbon footprint (CF) with source-wise breakdown, greenhouse gas (GHG) emission estimation, soil organic carbon (SOC) stocks, carbon sequestration rate (CSR), and global warming potential (GWP) using IPCC AR6 values.

Author(s)

Maintainer: Lalit Kumar Rolaniya <lalitrolaniya@gmail.com> (ORCID)

Authors:

- Hemant Poonia <pooniaheman80@gmail.com> (ORCID)
- Ram Lal Jat <khokharagro1986@gmail.com> (ORCID)
- Monika Punia <monikapunia02@gmail.com> (ORCID)
- Raja Ram Choudhary <rajaramchoudhary26@gmail.com>

See Also

Useful links:

- https://github.com/lalitrolaniya/wefnexus
- Report bugs at https://github.com/lalitrolaniya/wefnexus/issues

agronomic_efficiency *Agronomic Efficiency (AE)*

Description

Increase in economic yield per unit of nutrient applied.

Usage

```
agronomic_efficiency(
  yield_treated,
  yield_control,
  nutrient_applied,
  verbose = TRUE
)
```

Arguments

`yield_treated`	Numeric vector. Yield with nutrient applied (kg/ha).
`yield_control`	Numeric vector. Yield without nutrient, i.e., from control plot (kg/ha).
`nutrient_applied`	Numeric vector. Amount of nutrient applied (kg/ha).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

$$AE = \frac{Y_f - Y_0}{F}$$

Value

Numeric vector. Agronomic efficiency (kg grain per kg nutrient).

References

Dobermann, A. (2007). Nutrient use efficiency: measurement and management. In *Fertilizer Best Management Practices*, IFA, Paris, pp. 1-28. https://digitalcommons.unl.edu/agronomyfacpub/316/

Fixen, P. et al. (2015). Nutrient/fertilizer use efficiency: Measurement, current situation and trends. In *Managing Water and Fertilizer for Sustainable Agricultural Intensification*, IFA/IWMI/IPNI/IPI, pp. 8-38.

Congreves, K.A. et al. (2021). Nitrogen use efficiency definitions of today and tomorrow. *Frontiers in Plant Science*, 12, 637108. doi:10.3389/fpls.2021.637108

Examples

```
agronomic_efficiency(yield_treated = 4500, yield_control = 3000,
                     nutrient_applied = 120)
```

---

arid_pulse_nexus *Arid Pulse Nexus Dataset*

Description

A sample dataset with six conservation agriculture treatments in an arid pulse-based cropping system from western Rajasthan, India. Includes water, energy, food, nutrient, and carbon data for WEFNC nexus analysis.

Usage

```
arid_pulse_nexus
```

Format

A data frame with 6 rows and 26 variables:

treatment Character. Treatment combination name

tillage Character. Tillage method: CT (conventional), ZT (zero), PB (permanent beds)

irrigation Character. Irrigation method: Flood, Sprinkler, Drip, or SSDI (sub-surface drip)

residue Logical. Whether crop residue was retained

grain_yield Numeric. Grain yield (kg/ha)

straw_yield Numeric. Straw yield (kg/ha)

irrigation_applied Numeric. Irrigation water applied (mm)

effective_rainfall Numeric. Effective rainfall (mm)

total_water Numeric. Total water consumed (mm)

crop_et Numeric. Crop evapotranspiration (mm)

energy_input Numeric. Total energy input (MJ/ha)

energy_output_grain Numeric. Energy output from grain (MJ/ha)

energy_output_straw Numeric. Energy output from straw (MJ/ha)

n_applied Numeric. Nitrogen applied (kg/ha)

p_applied Numeric. Phosphorus applied (kg P2O5/ha)

k_applied Numeric. Potassium applied (kg K2O/ha)

n_uptake Numeric. Total plant nitrogen uptake (kg/ha)

p_uptake Numeric. Total plant phosphorus uptake (kg/ha)

grain_n_uptake Numeric. Grain nitrogen uptake (kg/ha)

diesel_use Numeric. Diesel consumption (L/ha)

electricity_kwh Numeric. Electricity consumption (kWh/ha)

soc_pct Numeric. Soil organic carbon (percent)

bulk_density Numeric. Soil bulk density (Mg/m3)

ghg_emission Numeric. Total GHG emission (kg CO2-eq/ha)

cost_cultivation Numeric. Cost of cultivation (INR/ha)

gross_return Numeric. Gross return (INR/ha)

Source

Simulated dataset based on typical experimental data from ICAR-Indian Institute of Pulses Research, Regional Centre, Bikaner, Rajasthan, India.

Examples

```
data(arid_pulse_nexus)
str(arid_pulse_nexus)
```

`caloric_yield` *Caloric Yield*

Description

Total caloric output per hectare from crop production.

Usage

```
caloric_yield(yield, caloric_value, verbose = TRUE)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`caloric_value`	Numeric vector. Caloric content (kcal/kg).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$CY = Yield \times CaloricValue$$

Value

Numeric vector. Caloric yield (kcal/ha).

Examples

```
caloric_yield(yield = 4500, caloric_value = 3400)
```

carbon_efficiency *Carbon Efficiency*

Description

Crop yield per unit of greenhouse gas emitted.

Usage

```
carbon_efficiency(yield, carbon_emission, verbose = TRUE)
```

Arguments

yield	Numeric vector. Crop yield (kg/ha).
carbon_emission	Numeric vector. Total GHG emission (kg CO2-eq/ha).
verbose	Logical. If TRUE, prints informational messages. Default TRUE.

Details

Formula:

$$CE = \frac{Yield}{CarbonFootprint}$$

Value

Numeric vector. Carbon efficiency (kg yield per kg CO2-eq). Higher values indicate more yield per unit emission.

Examples

```
carbon_efficiency(yield = 4500, carbon_emission = 2500)
```

carbon_footprint *Carbon Footprint of Crop Production*

Description

Estimates the carbon footprint from major emission sources in crop production, with source-wise breakdown and yield-scaled intensity. Uses IPCC AR6 default GWP values.

Usage

```
carbon_footprint(
  diesel_use = 0,
  electricity_use = 0,
  n_fertilizer = 0,
  p_fertilizer = 0,
  k_fertilizer = 0,
  pesticide_use = 0,
  seed_rate = 0,
  n2o_direct = NULL,
  ch4_emission = 0,
  yield = NULL,
  ef_diesel = 2.68,
  ef_electricity = 0.82,
  ef_n_manufacture = 4.96,
  ef_p_manufacture = 1.61,
  ef_k_manufacture = 0.57,
  ef_pesticide = 10.97,
  ef_seed = 0.58,
  gwp_n2o = 273,
  gwp_ch4 = 27,
  verbose = TRUE
)
```

Arguments

`diesel_use` Numeric. Diesel consumption (L/ha). Default 0.

`electricity_use`
Numeric. Electricity consumption (kWh/ha). Default 0.

`n_fertilizer` Numeric. Nitrogen fertilizer applied (kg N/ha). Default 0.

`p_fertilizer` Numeric. Phosphorus fertilizer applied (kg P2O5/ha). Default 0.

`k_fertilizer` Numeric. Potassium fertilizer applied (kg K2O/ha). Default 0.

`pesticide_use` Numeric. Pesticide active ingredient used (kg a.i./ha). Default 0.

`seed_rate` Numeric. Seed rate (kg/ha). Default 0.

`n2o_direct` Numeric. Direct N2O emissions from soil (kg N2O/ha). If `NULL` (default), estimated using IPCC Tier 1 default: 1 percent of applied N emitted as N2O-N, then converted to N2O using factor 44/28.

`ch4_emission` Numeric. Methane emission (kg CH4/ha). Default 0. Relevant for rice paddies.

`yield` Numeric. Crop yield (kg/ha) for intensity calculation. Default `NULL`.

`ef_diesel` Numeric. Emission factor for diesel (kg CO2/L). Default 2.68.

`ef_electricity` Numeric. Emission factor for grid electricity (kg CO2/kWh). Default 0.82 (India grid average, CEA 2023).

`ef_n_manufacture`
Numeric. Emission factor for N fertilizer manufacture (kg CO2-eq/kg N). Default 4.96 (urea-based).

`ef_p_manufacture`
: Numeric. Emission factor for P fertilizer manufacture (kg CO2-eq/kg P2O5). Default 1.61.

`ef_k_manufacture`
: Numeric. Emission factor for K fertilizer manufacture (kg CO2-eq/kg K2O). Default 0.57.

`ef_pesticide`
: Numeric. Emission factor for pesticide (kg CO2-eq/kg a.i.). Default 10.97.

`ef_seed`
: Numeric. Emission factor for seed production (kg CO2-eq/kg). Default 0.58.

`gwp_n2o`
: Numeric. Global warming potential of N2O relative to CO2. Default 273 (IPCC AR6, 100-year horizon).

`gwp_ch4`
: Numeric. Global warming potential of CH4 relative to CO2. Default 27 (IPCC AR6, 100-year horizon).

`verbose`
: Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$CF = \frac{\sum Activity_i \times EF_i}{Yield}$$

Value

A list with components:

total_cf Numeric. Total carbon footprint (kg CO2-eq/ha)

cf_intensity Numeric. Carbon footprint intensity (kg CO2-eq/kg yield), returned only when `yield` is provided and positive

breakdown Data frame with columns: source, emission_kg_CO2eq, and share_pct showing the contribution of each emission source

References

Lal, R. (2004). Carbon emission from farm operations. *Environment International*, 30(7), 981-990. doi:10.1016/j.envint.2004.01.005

Forster, P. et al. (2021). The Earth's energy budget, climate feedbacks, and climate sensitivity. In *Climate Change 2021: The Physical Science Basis* (IPCC AR6 WGI Chapter 7). doi:10.1017/9781009157896.009

IPCC (2019). *2019 Refinement to the 2006 IPCC Guidelines for National Greenhouse Gas Inventories*. Volume 4: Agriculture, Forestry and Other Land Use. ISBN:978-4-88788-232-4.

Examples

```
cf <- carbon_footprint(
  diesel_use = 60, electricity_use = 200,
  n_fertilizer = 120, p_fertilizer = 60, k_fertilizer = 40,
  pesticide_use = 1.5, seed_rate = 100, yield = 4500
)
cf$total_cf
```

```
cf$cf_intensity
cf$breakdown
```

---

carbon_sequestration_rate *Carbon Sequestration Rate*

Description

Annual rate of soil carbon change between two measurement time points.

Usage

```
carbon_sequestration_rate(soc_initial, soc_final, years, verbose = TRUE)
```

Arguments

soc_initial	Numeric vector. Initial SOC stock (Mg C/ha).
soc_final	Numeric vector. Final SOC stock (Mg C/ha).
years	Numeric. Number of years between measurements.
verbose	Logical. If TRUE, prints informational messages. Default TRUE.

Details

Formula:

$$CSR = \frac{SOC_{final} - SOC_{initial}}{Years}$$

Value

Numeric vector. Carbon sequestration rate (Mg C/ha/year). Positive values indicate net sequestration; negative values indicate net emission from soil.

References

Minasny, B. et al. (2017). Soil carbon 4 per mille. *Geoderma*, 292, 59-86. doi:10.1016/j.geoderma.2017.01.002

Examples

```
carbon_sequestration_rate(soc_initial = 28.5, soc_final = 31.2,
                          years = 5)
```

carbon_sustainability_index
Carbon Sustainability Index (CSI)

Description

Ratio of yield to net carbon emission, accounting for carbon sequestration by the soil.

Usage

```
carbon_sustainability_index(
  yield,
  carbon_emission,
  carbon_sequestered = 0,
  verbose = TRUE
)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`carbon_emission`	Numeric vector. Total GHG emission (kg CO2-eq/ha).
`carbon_sequestered`	Numeric vector. Carbon sequestered in soil (kg CO2-eq/ha). Default 0.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$CSI = \frac{Yield}{CarbonFootprint}$$

Value

Numeric vector. Carbon sustainability index. Higher values indicate better carbon sustainability. Returns `NA` where net emission is zero or negative (net sequestration exceeds emission).

References

Brentrup, F. et al. (2004). Environmental impact assessment of agricultural production systems using the life cycle assessment methodology. *European Journal of Agronomy*, 20(3), 247-264. doi:10.1016/S11610301(03)000248

Examples

```
carbon_sustainability_index(yield = 4500, carbon_emission = 2500,
                            carbon_sequestered = 500)
```

`crop_water_productivity` *Crop Water Productivity (CWP)*

Description

Computes crop water productivity as the ratio of yield to total evapotranspiration, analogous to the "more crop per drop" concept.

Usage

```
crop_water_productivity(yield, et, verbose = TRUE)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`et`	Numeric vector. Seasonal evapotranspiration (mm).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

$$CWP = \frac{Y}{ET \times 10}$$

The factor 10 converts mm to m3/ha.

Value

Numeric vector. Crop water productivity (kg/m3).

References

Zwart, S.J. & Bastiaanssen, W.G.M. (2004). Review of measured crop water productivity values for irrigated wheat, rice, cotton and maize. *Agricultural Water Management*, 69(2), 115-133. doi:10.1016/j.agwat.2004.04.007

Examples

```
crop_water_productivity(yield = 4500, et = 400)
```

crop_water_stress_index *Crop Water Stress Index (CWSI)*

Description

Computes the Crop Water Stress Index based on the ratio of actual to potential evapotranspiration.

Usage

```
crop_water_stress_index(actual_et, potential_et, verbose = TRUE)
```

Arguments

`actual_et`	Numeric vector. Actual evapotranspiration (mm).
`potential_et`	Numeric vector. Potential or reference evapotranspiration (mm).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

$$CWSI = 1 - \frac{ET_a}{ET_p}$$

Value

Numeric vector. CWSI values between 0 (no stress) and 1 (maximum stress). Values close to 0 indicate well-watered conditions; values close to 1 indicate severe water deficit.

References

Idso, S.B. et al. (1981). Normalizing the stress-degree-day parameter for environmental variability. *Agricultural Meteorology*, 24, 45-55. doi:10.1016/00021571(81)900327

Examples

```
crop_water_stress_index(c(4.5, 3.8, 3.0, 2.2), c(5.0, 5.0, 5.0, 5.0))
```

crop_yield_index *Crop Yield Index (CYI)*

Description

Relative yield performance compared to a check or control treatment.

Usage

```
crop_yield_index(yield, check_yield, verbose = TRUE)
```

Arguments

yield	Numeric vector. Treatment yields (kg/ha).
check_yield	Numeric. Control or check yield (kg/ha).
verbose	Logical. If TRUE, prints informational messages. Default TRUE.

Details

Formula:

$$CYI = \frac{Yield}{MaximumYield} \times 100$$

Value

Numeric vector. Yield index where 1.0 equals the check yield. Values above 1 indicate superiority over check.

Examples

```
crop_yield_index(yield = c(4500, 4200, 3800), check_yield = 4000)
```

depleted_fraction *Depleted Fraction (DF)*

Description

Ratio of water beneficially consumed (crop ET) to total water inflow, used to assess basin-level water allocation.

Usage

```
depleted_fraction(crop_et, total_inflow, verbose = TRUE)
```

Arguments

crop_et	Numeric vector. Beneficial crop evapotranspiration (mm).
total_inflow	Numeric vector. Total water inflow: irrigation plus effective rainfall (mm).
verbose	Logical. If TRUE, prints informational messages. Default TRUE.

Details

Formula:

$$DF = \frac{ET_{actual}}{Water_{supplied}}$$

Value

Numeric vector. Depleted fraction (proportion, 0 to 1).

References

Perry, C., Steduto, P., Allen, R.G. & Burt, C.M. (2009). Increasing productivity in irrigated agriculture: Agronomic constraints and hydrological realities. *Agricultural Water Management*, 96(11), 1517-1524. doi:10.1016/j.agwat.2009.05.005

Examples

```
depleted_fraction(crop_et = 320, total_inflow = 500)
```

---

energy_input *Total Energy Input*

Description

Computes total energy input for a cropping system from individual input components with energy equivalents.

Usage

```
energy_input(
  seed = 0,
  fertilizer_n = 0,
  fertilizer_p = 0,
  fertilizer_k = 0,
  fym = 0,
  pesticide = 0,
  diesel = 0,
  electricity = 0,
  human_labour = 0,
  machinery = 0,
```

```
  irrigation = 0,
  micronutrient = 0,
  biofertilizer = 0,
  solar_energy = 0,
  other = 0,
  verbose = TRUE
)
```

Arguments

`seed`	Numeric. Energy from seed (MJ/ha). Default 0.
`fertilizer_n`	Numeric. Energy from nitrogen fertilizer (MJ/ha). Default 0.
`fertilizer_p`	Numeric. Energy from phosphorus fertilizer (MJ/ha). Default 0.
`fertilizer_k`	Numeric. Energy from potassium fertilizer (MJ/ha). Default 0.
`fym`	Numeric. Energy from farmyard manure (MJ/ha). Default 0.
`pesticide`	Numeric. Energy from pesticides: herbicides, insecticides, fungicides (MJ/ha). Default 0.
`diesel`	Numeric. Energy from diesel fuel (MJ/ha). Default 0.
`electricity`	Numeric. Energy from electricity for irrigation (MJ/ha). Default 0.
`human_labour`	Numeric. Energy from human labour (MJ/ha). Default 0.
`machinery`	Numeric. Energy from machinery depreciation (MJ/ha). Default 0.
`irrigation`	Numeric. Energy for canal/pumped irrigation (MJ/ha). Default 0.
`micronutrient`	Numeric. Energy from micronutrient fertilizers (MJ/ha). Default 0.
`biofertilizer`	Numeric. Energy from biofertilizers (MJ/ha). Default 0.
`solar_energy`	Numeric. Energy from solar pumping systems (MJ/ha). Default 0.
`other`	Numeric. Any additional energy inputs (MJ/ha). Default 0.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$EI = \sum(Input_i \times EE_i)$$

Value

Numeric. Total energy input (MJ/ha), computed as the sum of all individual energy input components.

References

Mittal, V.K. & Dhawan, K.C. (1988). Research manual on energy requirements in agricultural sector. Punjab Agricultural University, Ludhiana, India.

Chaudhary, V.P. et al. (2009). Energy auditing of diversified rice-wheat cropping systems in the Indo-Gangetic Plains. *Energy*, 34(9), 1091-1096. doi:10.1016/j.energy.2009.04.017

Examples

```
energy_input(seed = 250, fertilizer_n = 3600, fertilizer_p = 500,
             diesel = 2800, human_labour = 180, machinery = 1200,
             irrigation = 1500)
```

---

energy_intensity *Energy Intensity*

Description

Energy consumed per unit of crop yield.

Usage

```
energy_intensity(energy_in, yield, verbose = TRUE)
```

Arguments

`energy_in`	Numeric vector. Total energy input (MJ/ha).
`yield`	Numeric vector. Crop yield (kg/ha).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$EI = \frac{EnergyInput}{Yield}$$

Value

Numeric vector. Energy intensity (MJ/kg). Lower values indicate more energy-efficient production.

Examples

```
energy_intensity(energy_in = 12000, yield = 4500)
```

`energy_output` *Total Energy Output*

Description

Computes total energy output from grain and straw/stover yields using crop-specific energy coefficients.

Usage

```
energy_output(
  grain_yield,
  straw_yield = 0,
  grain_energy_coeff = 14.7,
  straw_energy_coeff = 12.5,
  verbose = TRUE
)
```

Arguments

`grain_yield` Numeric vector. Grain or economic yield (kg/ha).

`straw_yield` Numeric vector. Straw, stover, or by-product yield (kg/ha). Default 0.

`grain_energy_coeff` Numeric. Energy coefficient for grain (MJ/kg). Default 14.7 (wheat grain equivalent).

`straw_energy_coeff` Numeric. Energy coefficient for straw (MJ/kg). Default 12.5 (wheat straw equivalent).

`verbose` Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

$$E_{out} = (Y_g \times C_g) + (Y_s \times C_s)$$

where Y_g and Y_s are grain and straw yields, and C_g and C_s are their respective energy coefficients.

Value

Numeric vector. Total energy output (MJ/ha).

References

Devasenapathy, P., Senthilkumar, G. & Shanmugam, P.M. (2009). Energy management in crop production. Indian Journal of Agronomy, 54(1), 80-90.

Examples

```
energy_output(grain_yield = 4500, straw_yield = 5500)
energy_output(grain_yield = 1500, straw_yield = 2000,
              grain_energy_coeff = 14.3, straw_energy_coeff = 12.5)
```

`energy_productivity` *Energy Productivity*

Description

Crop yield produced per unit of energy input.

Usage

```
energy_productivity(yield, energy_in, verbose = TRUE)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`energy_in`	Numeric vector. Total energy input (MJ/ha).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$EP = \frac{Yield}{EnergyInput}$$

Value

Numeric vector. Energy productivity (kg/MJ). Higher values indicate better yield per unit energy invested.

Examples

```
energy_productivity(yield = 4500, energy_in = 12000)
```

energy_profitability *Energy Profitability*

Description

Economic return per unit of energy invested.

Usage

```
energy_profitability(gross_return, energy_in, verbose = TRUE)
```

Arguments

gross_return	Numeric vector. Gross economic return (currency/ha).
energy_in	Numeric vector. Total energy input (MJ/ha).
verbose	Logical. If TRUE, prints informational messages. Default TRUE.

Details

Formula:

$$EPr = \frac{EnergyOutput - EnergyInput}{EnergyInput}$$

Value

Numeric vector. Energy profitability (currency/MJ).

Examples

```
energy_profitability(gross_return = 112500, energy_in = 12000)
```

energy_use_efficiency *Energy Use Efficiency (EUE)*

Description

Ratio of total energy output to total energy input.

Usage

```
energy_use_efficiency(energy_out, energy_in, verbose = TRUE)
```

Arguments

- `energy_out` Numeric vector. Total energy output (MJ/ha).
- `energy_in` Numeric vector. Total energy input (MJ/ha).
- `verbose` Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

$$EUE = \frac{E_{out}}{E_{in}}$$

Value

Numeric vector. Energy use efficiency (dimensionless ratio). Values greater than 1 indicate a positive energy balance.

Examples

```
energy_use_efficiency(energy_out = 135000, energy_in = 12000)
```

`eroi` *Energy Return on Investment (EROI)*

Description

Computes the Energy Return on Investment, a key metric for assessing whether an agricultural system produces more energy than it consumes. EROI is mathematically identical to energy use efficiency (EUE) but is the preferred term in energy and sustainability science literature.

Usage

```
eroi(energy_out, energy_in, include_solar = FALSE, verbose = TRUE)
```

Arguments

- `energy_out` Numeric vector. Total energy output from the agricultural system (MJ/ha), including grain, straw, and any co-products.
- `energy_in` Numeric vector. Total energy input invested in the agricultural system (MJ/ha), including all direct (diesel, electricity) and indirect (fertilizer manufacture, machinery depreciation) energy inputs.
- `include_solar` Logical. If `TRUE`, includes captured solar energy in the output. Default `FALSE` (standard practice).
- `verbose` Logical. If `TRUE`, prints informational messages and interpretation. Default `TRUE`.

Details

$$EROI = \frac{E_{out}}{E_{in}}$$

EROI is the fundamental metric of net energy analysis. In agricultural contexts, EROI typically ranges from 2 to 15 for conventional cropping systems, with higher values for low-input or conservation agriculture systems.

Value

Numeric vector. EROI values (dimensionless ratio). Interpreted as follows:

EROI > 5 Highly energy-profitable system

1 < EROI < 5 Energy-positive but moderate return

EROI = 1 Break-even: energy output equals input

EROI < 1 Energy sink: system consumes more than it produces

References

Hall, C.A.S., Lambert, J.G. & Balogh, S.B. (2014). EROI of different fuels and the implications for society. *Energy Policy*, 64, 141-152. doi:10.1016/j.enpol.2013.05.049

Murphy, D.J. & Hall, C.A.S. (2010). Year in review - EROI or energy return on (energy) invested. *Annals of the New York Academy of Sciences*, 1185(1), 102-118. doi:10.1111/j.1749-6632.2009.05282.x

Murphy, D.J. et al. (2022). Energy return on investment of major energy carriers: Review and harmonization. *Sustainability*, 14(12), 7098. doi:10.3390/su14127098

Examples

```
# Conservation agriculture with low input
eroi(energy_out = 59800, energy_in = 8500)

# Conventional tillage with high input
eroi(energy_out = 40800, energy_in = 12500)

# Multiple treatments
eroi(energy_out = c(40800, 50500, 59800),
     energy_in = c(12500, 9800, 8500))
```

`food_productivity_index` *Food Productivity Index (FPI)*

Description

Computes a composite food productivity index considering yield relative to potential, and optionally nutritional quality.

Usage

```
food_productivity_index(
  yield,
  reference_yield,
  protein_content = NULL,
  caloric_value = NULL,
  verbose = TRUE
)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`reference_yield`	Numeric. Reference or potential yield (kg/ha).
`protein_content`	Numeric vector. Protein content (percent). Default `NULL`.
`caloric_value`	Numeric vector. Caloric value (kcal/100g). Default `NULL`.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$FPI = \frac{Yield}{ReferenceYield} \times 100$$

Value

Numeric vector. Food productivity index on a 0 to 1 scale. When protein and caloric values are provided, weights are 0.50 for yield ratio, 0.25 for protein, and 0.25 for caloric content.

Examples

```
food_productivity_index(yield = 4500, reference_yield = 6000,
                        protein_content = 12.5, caloric_value = 340)
```

ghg_emission *GHG Emission Estimation from Field Operations*

Description

Quick estimation of greenhouse gas emissions (N2O, CH4, CO2) from agricultural field operations using IPCC Tier 1 default factors.

Usage

```
ghg_emission(
  n_applied,
  residue_burned = FALSE,
  residue_amount = 0,
  paddy_days = 0,
  tillage = "conventional",
  gwp_n2o = 273,
  gwp_ch4 = 27,
  verbose = TRUE
)
```

Arguments

`n_applied`	Numeric. N fertilizer applied (kg/ha).
`residue_burned`	Logical. Whether crop residues are burned. Default `FALSE`.
`residue_amount`	Numeric. Residue amount (kg/ha) if burned. Default 0.
`paddy_days`	Numeric. Days under flooded paddy conditions. Default 0.
`tillage`	Character. Tillage type: `"conventional"`, `"reduced"`, or `"zero"`. Default `"conventional"`.
`gwp_n2o`	Numeric. GWP for N2O. Default 273 (IPCC AR6).
`gwp_ch4`	Numeric. GWP for CH4. Default 27 (IPCC AR6).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$GHG = \sum(Activity_i \times EF_i)$$

where EF is the emission factor for each activity.

Value

A data frame with four columns:

source Character. Emission source description

gas Character. Greenhouse gas type

emission_kg Numeric. Raw emission (kg/ha)

CO2_eq_kg Numeric. Emission in CO2 equivalents (kg CO2-eq/ha)

References

IPCC (2019). *2019 Refinement to the 2006 IPCC Guidelines for National Greenhouse Gas Inventories*. Volume 4, Chapter 11: N2O emissions from managed soils, and CO2 emissions from lime and urea application. ISBN:978-4-88788-232-4.

Examples

```
ghg_emission(n_applied = 120, tillage = "zero")
ghg_emission(n_applied = 150, paddy_days = 90)
```

---

global_warming_potential
Global Warming Potential (GWP)

Description

Converts individual greenhouse gas emissions to CO2 equivalents using IPCC AR6 Global Warming Potential values.

Usage

```
global_warming_potential(
  co2 = 0,
  ch4 = 0,
  n2o = 0,
  gwp_ch4 = 27,
  gwp_n2o = 273,
  time_horizon = "100yr",
  verbose = TRUE
)
```

Arguments

`co2`	Numeric. CO2 emission (kg/ha). Default 0.
`ch4`	Numeric. CH4 emission (kg/ha). Default 0.
`n2o`	Numeric. N2O emission (kg/ha). Default 0.
`gwp_ch4`	Numeric. GWP of CH4. Default 27 (IPCC AR6, 100-year).
`gwp_n2o`	Numeric. GWP of N2O. Default 273 (IPCC AR6, 100-year).
`time_horizon`	Character. `"100yr"` (default) or `"20yr"` for different assessment periods.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$GWP = CO_2 + CH_4 \times 27 + N_2O \times 273$$

IPCC AR6 100-year GWP values (with climate-carbon feedbacks): CH4 = 27, N2O = 273.

IPCC AR6 20-year GWP values: CH4 = 81, N2O = 273.

Note: Previous IPCC AR5 values were CH4 = 34, N2O = 298 (without climate-carbon feedbacks) or CH4 = 28, N2O = 265 (without).

Value

Numeric. Total global warming potential (kg CO2-eq/ha).

References

Forster, P. et al. (2021). The Earth's energy budget, climate feedbacks, and climate sensitivity. In *Climate Change 2021: The Physical Science Basis* (IPCC AR6 WGI Chapter 7), Table 7.15. doi:10.1017/9781009157896.009

Examples

```
global_warming_potential(co2 = 500, ch4 = 50, n2o = 2)
global_warming_potential(co2 = 500, ch4 = 50, n2o = 2,
                         time_horizon = "20yr")
```

---

harvest_index *Harvest Index (HI)*

Description

Proportion of economic yield to total above-ground biological yield.

Usage

```
harvest_index(economic_yield, biological_yield, verbose = TRUE)
```

Arguments

economic_yield	Numeric vector. Grain or economic yield (kg/ha).
biological_yield	Numeric vector. Total above-ground biomass yield (kg/ha), which is grain plus straw/stover.
verbose	Logical. If TRUE, prints informational messages. Default TRUE.

Details

$$HI = \frac{Y_{econ}}{Y_{biol}}$$

Value

Numeric vector. Harvest index as a proportion (0 to 1). A warning is issued if any value exceeds 1.

References

Hay, R.K.M. (1995). Harvest index: A review of its use in plant breeding and crop physiology. *Annals of Applied Biology*, 126(1), 197-216. doi:10.1111/j.17447348.1995.tb05015.x

Examples

```
harvest_index(economic_yield = 4500, biological_yield = 10000)
```

`human_energy_profitability` *Human Energy Profitability*

Description

Ratio of energy output to human labour energy input, measuring the degree of human labour amplification by the farming system.

Usage

```
human_energy_profitability(energy_out, human_energy, verbose = TRUE)
```

Arguments

`energy_out`	Numeric vector. Total energy output (MJ/ha).
`human_energy`	Numeric vector. Human labour energy input (MJ/ha).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$HEP = \frac{EnergyOutput}{HumanEnergyInput}$$

Value

Numeric vector. Human energy profitability (dimensionless ratio).

Examples

```
human_energy_profitability(energy_out = 135000, human_energy = 180)
```

`internal_utilization_efficiency`

Internal Utilization Efficiency (IUE)

Description

Yield produced per unit of total nutrient in the plant.

Usage

```
internal_utilization_efficiency(yield, total_uptake, verbose = TRUE)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`total_uptake`	Numeric vector. Total plant nutrient uptake (kg/ha).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$IE = \frac{Yield}{TotalNutrientUptake}$$

Value

Numeric vector. Internal utilization efficiency (kg yield per kg nutrient uptake).

Examples

```
internal_utilization_efficiency(yield = 4500, total_uptake = 100)
```

irrigation_efficiency *Irrigation Efficiency*

Description

Computes multiple irrigation efficiency metrics: conveyance, application, overall, and consumptive use efficiency.

Usage

```
irrigation_efficiency(
  water_delivered,
  water_diverted,
  water_stored = NULL,
  crop_et = NULL,
  verbose = TRUE
)
```

Arguments

`water_delivered`	Numeric vector. Water delivered to field (mm or m3).
`water_diverted`	Numeric vector. Water diverted from source (mm or m3).
`water_stored`	Numeric vector. Water stored in root zone (mm or m3). If `NULL`, application and overall efficiency are not computed.
`crop_et`	Numeric vector. Crop evapotranspiration (mm). If `NULL`, consumptive use efficiency is not computed.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$IE = \frac{Water_{crop}}{Water_{applied}} \times 100$$

Value

A data frame with efficiency metrics expressed as percentages. Columns depend on inputs provided:

conveyance_eff Conveyance efficiency (percent), always computed

application_eff Application efficiency (percent), if `water_stored` is provided

overall_eff Overall efficiency (percent), if `water_stored` is provided

consumptive_eff Consumptive use efficiency (percent), if `crop_et` is provided

References

Grafton, R.Q. et al. (2018). The paradox of irrigation efficiency. *Science*, 361(6404), 748-750. doi:10.1126/science.aat9314

Examples

```
irrigation_efficiency(water_delivered = 400, water_diverted = 500,
                      water_stored = 350, crop_et = 320)
```

---

land_equivalent_ratio *Land Equivalent Ratio (LER)*

Description

Evaluates the productivity advantage of intercropping systems over sole cropping.

Usage

```
land_equivalent_ratio(
  yield_inter_a,
  yield_sole_a,
  yield_inter_b,
  yield_sole_b,
  verbose = TRUE
)
```

Arguments

`yield_inter_a`	Numeric vector. Yield of crop A in intercrop (kg/ha).
`yield_sole_a`	Numeric vector. Yield of crop A in sole crop (kg/ha).
`yield_inter_b`	Numeric vector. Yield of crop B in intercrop (kg/ha).
`yield_sole_b`	Numeric vector. Yield of crop B in sole crop (kg/ha).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

$$LER = \frac{Y_{iA}}{Y_{sA}} + \frac{Y_{iB}}{Y_{sB}}$$

Value

Numeric vector. LER values. Values greater than 1 indicate an intercropping advantage; less than 1 indicates a disadvantage.

References

Mead, R. & Willey, R.W. (1980). The concept of a land equivalent ratio and advantages in yields from intercropping. *Experimental Agriculture*, 16(3), 217-228. doi:10.1017/S0014479700010978

Examples

```
land_equivalent_ratio(yield_inter_a = 3500, yield_sole_a = 4500,
                      yield_inter_b = 800, yield_sole_b = 1200)
```

`net_energy` *Net Energy*

Description

Difference between total energy output and total energy input.

Usage

```
net_energy(energy_out, energy_in, verbose = TRUE)
```

Arguments

`energy_out`	Numeric vector. Total energy output (MJ/ha).
`energy_in`	Numeric vector. Total energy input (MJ/ha).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$NE = EnergyOutput - EnergyInput$$

Value

Numeric vector. Net energy (MJ/ha). Positive values indicate net energy gain; negative values indicate net energy loss.

Examples

```
net_energy(energy_out = 135000, energy_in = 12000)
```

nexus_heatmap — *Nexus Heatmap*

Description

Creates a heatmap showing nexus dimension scores across treatments.

Usage

```
nexus_heatmap(
  scores,
  row_labels = NULL,
  col_labels = NULL,
  title = "WEFNC Nexus Heatmap",
  color_palette = NULL
)
```

Arguments

scores	A matrix or data frame where rows represent treatments and columns represent nexus dimensions.
row_labels	Character vector. Treatment names.
col_labels	Character vector. Dimension names.
title	Character. Plot title. Default "WEFNC Nexus Heatmap".
color_palette	Character vector of colors for gradient. Default is a red-yellow-green palette.

Value

Invisibly returns the input scores matrix. Called primarily for the side effect of generating a heatmap plot.

Examples

```
scores <- matrix(c(0.85, 0.70, 0.90, 0.65, 0.80,
                    0.72, 0.80, 0.85, 0.70, 0.60),
                 nrow = 2, byrow = TRUE)
nexus_heatmap(scores,
  row_labels = c("Conservation", "Conventional"),
  col_labels = c("Water", "Energy", "Food", "Nutrient", "Carbon"))
```

nexus_heatmap_gg *Nexus Heatmap (ggplot2)*

Description

Publication-quality heatmap using ggplot2 for WEFNC nexus scores.

Usage

```
nexus_heatmap_gg(
  scores,
  treatment_names = NULL,
  dim_labels = c("Water", "Energy", "Food", "Nutrient", "Carbon"),
  title = "WEFNC Nexus Heatmap"
)
```

Arguments

- `scores` A matrix or data frame (treatments as rows, dimensions as columns).
- `treatment_names` Character vector. Row labels.
- `dim_labels` Character vector. Column labels. Default: Water, Energy, Food, Nutrient, Carbon.
- `title` Character. Plot title.

Value

A ggplot2 object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  scores <- matrix(c(0.85, 0.70, 0.90, 0.65, 0.80,
                     0.72, 0.80, 0.85, 0.70, 0.60),
                   nrow = 2, byrow = TRUE)
  nexus_heatmap_gg(scores, treatment_names = c("CA", "CT"))
}
```

`nexus_index` *WEFNC Nexus Index*

Description

Computes a weighted composite Water-Energy-Food-Nutrient-Carbon nexus sustainability index from normalized dimension scores.

Usage

```
nexus_index(
  water_score,
  energy_score,
  food_score,
  nutrient_score,
  carbon_score,
  weights = rep(0.2, 5),
  verbose = TRUE
)
```

Arguments

`water_score`	Numeric vector. Water performance score (0 to 1).
`energy_score`	Numeric vector. Energy performance score (0 to 1).
`food_score`	Numeric vector. Food productivity score (0 to 1).
`nutrient_score`	Numeric vector. Nutrient use efficiency score (0 to 1).
`carbon_score`	Numeric vector. Carbon performance score (0 to 1), where higher values indicate lower carbon footprint.
`weights`	Numeric vector of length 5. Weights for Water, Energy, Food, Nutrient, and Carbon dimensions. Must sum to 1. Default is equal weighting (0.2 each).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

$$NI = w_W S_W + w_E S_E + w_F S_F + w_N S_N + w_C S_C$$

Value

Numeric vector. Composite nexus index on a 0 to 1 scale, where higher values indicate better overall sustainability.

Examples

```
nexus_index(water_score = c(0.85, 0.72), energy_score = c(0.70, 0.80),
            food_score = c(0.90, 0.85), nutrient_score = c(0.65, 0.70),
            carbon_score = c(0.80, 0.60))
```

---

nexus_radar *Nexus Radar Plot*

Description

Creates a radar (spider/web) chart showing the WEFNC nexus profile of one or more treatments.

Usage

```
nexus_radar(
  scores,
  labels = c("Water", "Energy", "Food", "Nutrient", "Carbon"),
  treatment_names = NULL,
  colors = NULL,
  title = "WEFNC Nexus Profile",
  fill = TRUE,
  alpha = 0.2
)
```

Arguments

`scores`	A matrix or data frame where rows represent treatments and columns represent nexus dimensions. Values should be on a 0 to 1 scale.
`labels`	Character vector. Names for nexus dimensions. Default: Water, Energy, Food, Nutrient, Carbon.
`treatment_names`	Character vector. Labels for each treatment.
`colors`	Character vector. Colors for each treatment.
`title`	Character. Plot title. Default "WEFNC Nexus Profile".
`fill`	Logical. If `TRUE`, fills the radar polygon area. Default `TRUE`.
`alpha`	Numeric. Fill transparency (0 to 1). Default 0.2.

Value

Invisibly returns the input `scores` matrix. Called primarily for the side effect of generating a radar plot.

Examples

```
scores <- matrix(c(0.85, 0.70, 0.90, 0.65, 0.80,
                    0.72, 0.80, 0.85, 0.70, 0.60),
                 nrow = 2, byrow = TRUE)
nexus_radar(scores,
  treatment_names = c("CA+SSDI", "CT+Flood"))
```

---

nexus_radar_gg *Nexus Radar Plot (ggplot2)*

Description

Publication-quality radar chart using ggplot2 for WEFNC nexus profiles.

Usage

```
nexus_radar_gg(
  scores,
  treatment_names = NULL,
  dim_labels = c("Water", "Energy", "Food", "Nutrient", "Carbon"),
  title = "WEFNC Nexus Profile"
)
```

Arguments

`scores`	A matrix or data frame (treatments as rows, dimensions as columns). Values should be on a 0 to 1 scale.
`treatment_names`	Character vector. Labels for treatments.
`dim_labels`	Character vector. Labels for nexus dimensions. Default: Water, Energy, Food, Nutrient, Carbon.
`title`	Character. Plot title.

Value

A ggplot2 object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  scores <- matrix(c(0.85, 0.70, 0.90, 0.65, 0.80,
                      0.72, 0.80, 0.85, 0.70, 0.60),
                   nrow = 2, byrow = TRUE)
  nexus_radar_gg(scores, treatment_names = c("CA", "CT"))
}
```

`nexus_sensitivity` *Nexus Weight Sensitivity Analysis*

Description

Evaluates how the composite nexus index changes as the weight of one dimension varies from 0 to its maximum feasible value, with remaining weight distributed equally among the other four dimensions.

Usage

```
nexus_sensitivity(
  water_score,
  energy_score,
  food_score,
  nutrient_score,
  carbon_score,
  treatment_names = NULL,
  steps = 20,
  verbose = TRUE
)
```

Arguments

`water_score`	Numeric vector. Water scores (0 to 1).
`energy_score`	Numeric vector. Energy scores (0 to 1).
`food_score`	Numeric vector. Food scores (0 to 1).
`nutrient_score`	Numeric vector. Nutrient scores (0 to 1).
`carbon_score`	Numeric vector. Carbon scores (0 to 1).
`treatment_names`	Character vector. Treatment labels.
`steps`	Integer. Number of weight steps to evaluate. Default 20.
`verbose`	Logical. If `TRUE`, prints a message. Default `TRUE`.

Value

A data frame with columns: dimension, weight, treatment, and nexus_index showing how the index changes as each dimension's weight varies.

Examples

```
sa <- nexus_sensitivity(
  water_score = c(0.9, 0.5), energy_score = c(0.6, 0.8),
  food_score = c(0.8, 0.7), nutrient_score = c(0.7, 0.6),
  carbon_score = c(0.5, 0.9),
  treatment_names = c("CA", "CT"), steps = 10
```

```
)
head(sa)
```

---

nexus_summary *Nexus Summary*

Description

Generates a comprehensive one-call nexus analysis from raw agronomic field data, computing all dimension metrics, normalizing, and producing the composite index.

Usage

```
nexus_summary(
  yield,
  water_consumed,
  energy_input,
  energy_output,
  n_applied,
  n_uptake,
  carbon_emission,
  treatment_names = NULL,
  weights = rep(0.2, 5),
  verbose = TRUE
)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`water_consumed`	Numeric vector. Total water consumed (mm).
`energy_input`	Numeric vector. Total energy input (MJ/ha).
`energy_output`	Numeric vector. Total energy output (MJ/ha).
`n_applied`	Numeric vector. Nitrogen applied (kg/ha).
`n_uptake`	Numeric vector. Nitrogen total uptake (kg/ha).
`carbon_emission`	Numeric vector. Total GHG emission (kg CO2-eq/ha).
`treatment_names`	Character vector. Treatment labels. Default `NULL`.
`weights`	Numeric vector of length 5. Dimension weights. Default equal weights.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Value

A data frame with one row per treatment and columns:

treatment Character. Treatment name

yield_kg_ha Numeric. Crop yield

WUE_kg_mm Numeric. Water use efficiency

EUE_ratio Numeric. Energy use efficiency ratio

EROI Numeric. Energy return on investment

net_energy_MJ Numeric. Net energy balance

PFP_kg_kgN Numeric. Partial factor productivity of N

carbon_eff Numeric. Carbon efficiency

carbon_intensity Numeric. Carbon intensity

W_score Numeric. Normalized water score

E_score Numeric. Normalized energy score

F_score Numeric. Normalized food score

N_score Numeric. Normalized nutrient score

C_score Numeric. Normalized carbon score

nexus_index Numeric. Composite nexus index

Examples

```
nexus_summary(
  yield = c(4500, 4200, 3800),
  water_consumed = c(450, 400, 350),
  energy_input = c(12000, 11000, 9500),
  energy_output = c(135000, 125000, 112000),
  n_applied = c(120, 120, 120),
  n_uptake = c(100, 90, 80),
  carbon_emission = c(2500, 2200, 1800),
  treatment_names = c("CA+Drip", "CT+Sprinkler", "ZT+Rainfed")
)
```

---

nexus_sustainability_score

Nexus Sustainability Score

Description

Comprehensive sustainability score with categorization based on user-defined thresholds.

Usage

```
nexus_sustainability_score(
  water_score,
  energy_score,
  food_score,
  nutrient_score,
  carbon_score,
  weights = rep(0.2, 5),
  thresholds = c(0.4, 0.6, 0.8),
  verbose = TRUE
)
```

Arguments

`water_score`	Numeric vector. Water dimension score (0 to 1).
`energy_score`	Numeric vector. Energy dimension score (0 to 1).
`food_score`	Numeric vector. Food dimension score (0 to 1).
`nutrient_score`	Numeric vector. Nutrient dimension score (0 to 1).
`carbon_score`	Numeric vector. Carbon dimension score (0 to 1).
`weights`	Numeric vector of length 5. Default equal weights.
`thresholds`	Numeric vector of length 3. Breakpoints for sustainability categories: c(low, medium, high). Default c(0.4, 0.6, 0.8).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$NSS = \frac{1}{n} \sum_{i=1}^{n} S_i$$

where S_i are normalized sub-dimension scores.

Value

A data frame with columns:

nexus_score Numeric. Composite sustainability score (0 to 1)

category Character. Sustainability category: Unsustainable, Low Sustainability, Moderately Sustainable, or Highly Sustainable

water Numeric. Water dimension input score

energy Numeric. Energy dimension input score

food Numeric. Food dimension input score

nutrient Numeric. Nutrient dimension input score

carbon Numeric. Carbon dimension input score

Examples

```
nexus_sustainability_score(
  water_score = c(0.85, 0.60, 0.45),
  energy_score = c(0.70, 0.55, 0.35),
  food_score = c(0.90, 0.80, 0.50),
  nutrient_score = c(0.65, 0.70, 0.40),
  carbon_score = c(0.80, 0.50, 0.30)
)
```

---

nexus_tradeoff *Nexus Trade-off Matrix*

Description

Computes pairwise correlation matrix among nexus dimensions to identify synergies (positive) and trade-offs (negative).

Usage

```
nexus_tradeoff(data, method = "pearson", verbose = TRUE)
```

Arguments

`data`	A data frame with numeric columns for each nexus dimension (e.g., WUE, EUE, yield, NUE, carbon efficiency).
`method`	Character. Correlation method: `"pearson"` (default), `"spearman"`, or `"kendall"`.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Value

A numeric correlation matrix with pairwise correlations. Positive values suggest synergies; negative values suggest trade-offs between dimensions.

Examples

```
df <- data.frame(WUE = c(8.5, 7.2, 6.5, 9.0, 7.8),
                 EUE = c(11.2, 12.5, 9.8, 10.5, 11.0),
                 Yield = c(1500, 1380, 1250, 1650, 1580),
                 NUE = c(25, 23, 20.8, 27.5, 26.3),
                 CF = c(2500, 3000, 1800, 2800, 2200))
nexus_tradeoff(df)
```

`normalize_minmax` *Normalize Data (Min-Max)*

Description

Min-max normalization to scale values between 0 and 1.

Usage

```
normalize_minmax(x, inverse = FALSE)
```

Arguments

`x`	Numeric vector.
`inverse`	Logical. If `TRUE`, higher original values receive lower normalized scores. Use for metrics where lower is better, such as carbon footprint or energy intensity. Default `FALSE`.

Details

Formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Value

Numeric vector normalized to the 0 to 1 range. Returns 0.5 for all elements if the input has zero range.

Examples

```
normalize_minmax(c(10, 20, 30, 40, 50))
normalize_minmax(c(10, 20, 30, 40, 50), inverse = TRUE)
```

`normalize_zscore` *Normalize Data (Z-score)*

Description

Standardization to z-scores (mean = 0, standard deviation = 1).

Usage

```
normalize_zscore(x)
```

Arguments

`x`	Numeric vector.

Details

Formula:

$$Z = \frac{X - \mu}{\sigma}$$

Value

Numeric vector of z-scores. Returns all zeros if standard deviation is zero.

Examples

```
normalize_zscore(c(10, 20, 30, 40, 50))
```

---

nutrient_balance *Nutrient Balance Sheet*

Description

Input-output nutrient balance for major nutrients (N, P, K).

Usage

```
nutrient_balance(
  n_input,
  p_input,
  k_input,
  n_output,
  p_output,
  k_output,
  verbose = TRUE
)
```

Arguments

`n_input`	Numeric. Total N input from all sources (kg/ha).
`p_input`	Numeric. Total P input (kg/ha).
`k_input`	Numeric. Total K input (kg/ha).
`n_output`	Numeric. Total N removal or uptake (kg/ha).
`p_output`	Numeric. Total P removal or uptake (kg/ha).
`k_output`	Numeric. Total K removal or uptake (kg/ha).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$NB = NutrientApplied - NutrientRemoved$$

Value

A data frame with five columns:

nutrient Character. Nutrient name: N, P, or K

input_kg_ha Numeric. Total nutrient input (kg/ha)

output_kg_ha Numeric. Total nutrient output (kg/ha)

balance_kg_ha Numeric. Balance: input minus output (kg/ha). Positive = surplus; negative = depletion

output_input_ratio Numeric. Output to input ratio

Examples

```
nutrient_balance(n_input = 120, p_input = 60, k_input = 40,
                 n_output = 95, p_output = 25, k_output = 80)
```

`nutrient_harvest_index` *Nutrient Harvest Index (NHI)*

Description

Proportion of total plant nutrient partitioned to grain.

Usage

```
nutrient_harvest_index(
  grain_nutrient_uptake,
  total_nutrient_uptake,
  verbose = TRUE
)
```

Arguments

`grain_nutrient_uptake`
Numeric vector. Nutrient uptake in grain (kg/ha).

`total_nutrient_uptake`
Numeric vector. Total plant nutrient uptake (kg/ha).

`verbose` Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$HI = \frac{EconomicYield}{BiologicalYield}$$

Value

Numeric vector. Nutrient harvest index (proportion, 0 to 1).

Examples

```
nutrient_harvest_index(grain_nutrient_uptake = 75,
                       total_nutrient_uptake = 100)
```

---

nutrient_use_efficiency

Comprehensive Nutrient Use Efficiency (NUE)

Description

Computes all major nutrient use efficiency metrics for a given nutrient in a single call.

Usage

```
nutrient_use_efficiency(
  yield_treated,
  yield_control,
  nutrient_applied,
  uptake_treated,
  uptake_control,
  nutrient_name = "N",
  verbose = TRUE
)
```

Arguments

`yield_treated`	Numeric. Yield with nutrient applied (kg/ha).
`yield_control`	Numeric. Yield without nutrient (kg/ha).
`nutrient_applied`	Numeric. Nutrient applied (kg/ha).
`uptake_treated`	Numeric. Total nutrient uptake with fertilizer (kg/ha).
`uptake_control`	Numeric. Total nutrient uptake without fertilizer (kg/ha).
`nutrient_name`	Character. Name of the nutrient for labeling (e.g., "N", "P", "K"). Default "N".
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$NUE = \frac{Yield}{NutrientApplied}$$

Value

A data frame with one row and seven columns:

nutrient Character. Nutrient name

agronomic_eff Numeric. Agronomic efficiency (kg/kg)

physiological_eff Numeric. Physiological efficiency (kg/kg)

recovery_eff Numeric. Recovery efficiency (proportion)

partial_factor_prod Numeric. Partial factor productivity (kg/kg)

internal_util_eff Numeric. Internal utilization efficiency (kg/kg)

nutrient_harvest_idx Numeric. Estimated nutrient harvest index (proportion, based on 0.75 grain-to-total ratio assumption)

References

Dobermann, A. (2007). Nutrient use efficiency: measurement and management. In *Fertilizer Best Management Practices*, IFA, Paris, pp. 1-28. `https://digitalcommons.unl.edu/agronomyfacpub/316/`

Congreves, K.A. et al. (2021). Nitrogen use efficiency definitions of today and tomorrow. *Frontiers in Plant Science*, 12, 637108. doi:10.3389/fpls.2021.637108

Examples

```
nutrient_use_efficiency(
  yield_treated = 4500, yield_control = 3000,
  nutrient_applied = 120, uptake_treated = 100,
  uptake_control = 60, nutrient_name = "N"
)
```

`partial_factor_productivity`

Partial Factor Productivity (PFP)

Description

Total yield per unit of nutrient applied, without deduction of control yield.

Usage

```
partial_factor_productivity(yield, nutrient_applied, verbose = TRUE)
```

Arguments

yield	Numeric vector. Crop yield (kg/ha).
nutrient_applied	Numeric vector. Nutrient applied (kg/ha).
verbose	Logical. If TRUE, prints informational messages. Default TRUE.

Details

$$PFP = \frac{Y}{F}$$

Value

Numeric vector. Partial factor productivity (kg yield per kg nutrient applied).

Examples

```
partial_factor_productivity(yield = 4500, nutrient_applied = 120)
```

---

physiological_efficiency

Physiological Efficiency (PE)

Description

Yield increase per unit of nutrient uptake increase.

Usage

```
physiological_efficiency(
  yield_treated,
  yield_control,
  uptake_treated,
  uptake_control,
  verbose = TRUE
)
```

Arguments

yield_treated	Numeric vector. Yield with nutrient applied (kg/ha).
yield_control	Numeric vector. Yield without nutrient (kg/ha).
uptake_treated	Numeric vector. Total nutrient uptake with fertilizer (kg/ha).
uptake_control	Numeric vector. Total nutrient uptake without fertilizer (kg/ha).
verbose	Logical. If TRUE, prints informational messages. Default TRUE.

Details

$$PE = \frac{Y_f - Y_0}{U_f - U_0}$$

Value

Numeric vector. Physiological efficiency (kg grain per kg nutrient uptake). Returns `NA` where uptake difference is zero.

Examples

```
physiological_efficiency(yield_treated = 4500, yield_control = 3000,
                         uptake_treated = 100, uptake_control = 60)
```

`plot.cf_result` *Plot method for cf_result*

Description

Plot method for cf_result

Usage

```
## S3 method for class 'cf_result'
plot(x, ...)
```

Arguments

`x`	An object of class `cf_result`.
`...`	Additional arguments passed to `barplot`.

Value

No return value, called for the side effect of generating a horizontal bar plot showing carbon footprint by emission source.

plot.nexus_result — *Plot method for nexus_result*

Description

Plot method for nexus_result

Usage

```
## S3 method for class 'nexus_result'
plot(x, type = c("radar", "heatmap"), ...)
```

Arguments

`x`	An object of class `nexus_result`.
`type`	Character. Plot type: `"radar"` (default) or `"heatmap"`.
`...`	Additional arguments passed to `nexus_radar` or `nexus_heatmap`.

Value

Invisibly returns the scores matrix. Called for the side effect of generating a radar or heatmap plot.

plot_sensitivity — *Plot Nexus Sensitivity Analysis (ggplot2)*

Description

Visualises the output of `nexus_sensitivity` as faceted line plots showing how the nexus index changes with dimension weights.

Usage

```
plot_sensitivity(sa, title = "Nexus Weight Sensitivity")
```

Arguments

`sa`	Data frame returned by `nexus_sensitivity`.
`title`	Character. Plot title.

Value

A ggplot2 object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  sa <- nexus_sensitivity(
    water_score = c(0.9, 0.5), energy_score = c(0.6, 0.8),
    food_score = c(0.8, 0.7), nutrient_score = c(0.7, 0.6),
    carbon_score = c(0.5, 0.9),
    treatment_names = c("CA", "CT"), steps = 10
  )
  plot_sensitivity(sa)
}
```

---

print.cf_result *Print method for cf_result*

Description

Print method for cf_result

Usage

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

Arguments

x	An object of class `cf_result`.
...	Additional arguments passed to `print`.

Value

Invisibly returns the input object `x`. Called for the side effect of printing a formatted carbon footprint summary to the console.

---

print.nexus_result *Print method for nexus_result*

Description

Print method for nexus_result

Usage

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

Arguments

`x`	An object of class `nexus_result`.
`...`	Additional arguments passed to `print.data.frame`.

Value

Invisibly returns the input object `x`. Called for the side effect of printing a formatted nexus analysis summary to the console.

---

print.sustainability_result
Print method for sustainability_result

Description

Print method for sustainability_result

Usage

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

Arguments

`x`	An object of class `sustainability_result`.
`...`	Additional arguments passed to `print.data.frame`.

Value

Invisibly returns the input object `x`. Called for the side effect of printing sustainability scores and categories to the console.

---

production_efficiency_index
Production Efficiency Index (PEI)

Description

Yield produced per unit of production cost, measuring economic efficiency of crop production.

Usage

```
production_efficiency_index(yield, cost, verbose = TRUE)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`cost`	Numeric vector. Total cost of production (currency/ha).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$PEI = \frac{Yield}{TotalInputCost} \times 100$$

Value

Numeric vector. Production efficiency index (kg/currency unit).

Examples

```
production_efficiency_index(yield = 4500, cost = 28500)
```

`protein_yield` *Protein Yield*

Description

Total protein output per hectare from crop production.

Usage

```
protein_yield(yield, protein_content, verbose = TRUE)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`protein_content`	Numeric vector. Protein content (percent).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$PY = Yield \times ProteinContent/100$$

Value

Numeric vector. Protein yield (kg/ha).

Examples

```
protein_yield(yield = 4500, protein_content = 12.5)
```

`recovery_efficiency` *Recovery Efficiency (RE)*

Description

Proportion of applied nutrient that is recovered in crop biomass.

Usage

```
recovery_efficiency(
  uptake_treated,
  uptake_control,
  nutrient_applied,
  verbose = TRUE
)
```

Arguments

`uptake_treated`	Numeric vector. Total nutrient uptake with fertilizer (kg/ha).
`uptake_control`	Numeric vector. Total nutrient uptake without fertilizer (kg/ha).
`nutrient_applied`	Numeric vector. Nutrient applied (kg/ha).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

$$RE = \frac{U_f - U_0}{F}$$

Value

Numeric vector. Recovery efficiency as a proportion (0 to 1, but can exceed 1 due to priming effects or added benefits).

Examples

```
recovery_efficiency(uptake_treated = 100, uptake_control = 60,
                    nutrient_applied = 120)
```

soil_carbon_stock *Soil Organic Carbon Stock (SOC)*

Description

Estimates soil organic carbon stock for a given soil depth.

Usage

```
soil_carbon_stock(
  soc_pct,
  bulk_density,
  depth = 30,
  coarse_fraction = 0,
  verbose = TRUE
)
```

Arguments

`soc_pct`	Numeric vector. Soil organic carbon content (percent).
`bulk_density`	Numeric vector. Soil bulk density (Mg/m3 or g/cm3).
`depth`	Numeric. Soil sampling depth (cm). Default 30.
`coarse_fraction`	Numeric vector. Volume fraction of coarse fragments greater than 2 mm (0 to 1). Default 0.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

$$SOC = SOC_{pct} \times BD \times D \times (1 - CF) \times 0.1$$

where BD is bulk density (Mg/m3), D is depth (cm), CF is coarse fraction, and 0.1 is the conversion factor from (percent * Mg/m3 * cm) to Mg/ha.

Value

Numeric vector. SOC stock (Mg C/ha, equivalent to t C/ha).

References

Batjes, N.H. (1996). Total carbon and nitrogen in the soils of the world. *European Journal of Soil Science*, 47(2), 151-163. doi:10.1111/j.13652389.1996.tb01386.x

Examples

```
soil_carbon_stock(soc_pct = 0.65, bulk_density = 1.45, depth = 30)
```

specific_energy — *Specific Energy*

Description

Energy input per unit area per unit time, useful for comparing cropping systems of different durations.

Usage

```
specific_energy(energy_in, duration, verbose = TRUE)
```

Arguments

`energy_in`	Numeric vector. Total energy input (MJ/ha).
`duration`	Numeric vector. Crop duration (days).
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Formula:

$$SE = \frac{EnergyInput}{Area}$$

Value

Numeric vector. Specific energy (MJ/ha/day).

Examples

```
specific_energy(energy_in = 12000, duration = 150)
```

summary.nexus_result — *Summary method for nexus_result*

Description

Summary method for nexus_result

Usage

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

Arguments

object	An object of class nexus_result.
...	Additional arguments (currently unused).

Value

Invisibly returns the input object object. Called for the side effect of printing a detailed dimension-wise summary to the console.

---

system_productivity_index
System Productivity Index (SPI)

Description

Converts yields from a cropping system to a common crop-equivalent yield based on economic value.

Usage

```
system_productivity_index(yields, prices, base_price, verbose = TRUE)
```

Arguments

yields	Numeric vector. Yields of individual crops in the system (kg/ha).
prices	Numeric vector. Market prices of individual crops (currency/kg). Must be the same length as yields.
base_price	Numeric. Price of the base or reference crop (currency/kg).
verbose	Logical. If TRUE, prints informational messages. Default TRUE.

Details

$$SPI = \frac{\sum(Y_i \times P_i)}{P_{base}}$$

Value

Numeric. System productivity expressed as base crop equivalent yield (kg/ha).

Examples

```
system_productivity_index(yields = c(5000, 4500),
                          prices = c(22, 25), base_price = 25)
```

`water_footprint` *Water Footprint (WF)*

Description

Computes the water footprint of crop production decomposed into green, blue, and grey components following the Water Footprint Assessment framework.

Usage

```
water_footprint(
  green_water,
  blue_water,
  grey_water = 0,
  yield,
  area = 1,
  per_ton = FALSE,
  verbose = TRUE
)
```

Arguments

`green_water`	Numeric vector. Green water use, i.e., effective rainfall consumed by the crop during growth (mm).
`blue_water`	Numeric vector. Blue water use, i.e., irrigation water applied and consumed (mm).
`grey_water`	Numeric vector. Grey water, i.e., freshwater volume required to dilute pollutant load to acceptable standards (mm). Default is 0.
`yield`	Numeric vector. Crop yield (kg/ha).
`area`	Numeric. Crop area in hectares. Default is 1.
`per_ton`	Logical. If `TRUE`, returns water footprint in m3/ton. Default `FALSE`, returning m3/kg.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

The water footprint per unit mass of product is:

$$WF = \frac{CWU \times 10}{Y}$$

where CWU is crop water use (mm) and Y is yield (kg/ha). The factor 10 converts mm over 1 ha to m3.

Value

A data frame with four numeric columns:

green_wf Green water footprint (m3/kg or m3/ton)

blue_wf Blue water footprint (m3/kg or m3/ton)

grey_wf Grey water footprint (m3/kg or m3/ton)

total_wf Total water footprint (m3/kg or m3/ton)

References

Hoekstra, A.Y., Chapagain, A.K., Aldaya, M.M. & Mekonnen, M.M. (2011). *The Water Footprint Assessment Manual: Setting the Global Standard*. Earthscan, London. ISBN:9781849712798.

Mialyk, O. et al. (2024). Water footprints and crop water use of 175 individual crops for 1990-2019 simulated with a global crop model. *Scientific Data*, 11, 200. doi:10.1038/s41597024030513

Examples

```
water_footprint(green_water = 300, blue_water = 200,
                grey_water = 50, yield = 4000)
```

---

water_productivity *Water Productivity (WP)*

Description

Computes physical or economic water productivity per unit of water used.

Usage

```
water_productivity(
  yield,
  water_applied,
  price = NULL,
  unit = "mm",
  verbose = TRUE
)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`water_applied`	Numeric vector. Total water applied (mm or m3/ha).
`price`	Numeric. Market price per kg of produce (currency/kg). If `NULL` (default), returns physical water productivity.
`unit`	Character. Unit of water input: `"mm"` (default) or `"m3"`.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Physical water productivity:

$$WP = \frac{Y}{W_{m3}}$$

Economic water productivity:

$$WP_{econ} = \frac{Y \times P}{W_{m3}}$$

When `unit = "mm"`, water is converted to m3/ha using 1 mm = 10 m3/ha.

Value

Numeric vector. Physical water productivity in kg/m3 when `price` is `NULL`, or economic water productivity in currency/m3 when `price` is specified.

References

Molden, D. et al. (2010). Improving agricultural water productivity: Between optimism and caution. *Agricultural Water Management*, 97(4), 528-535. doi:10.1016/j.agwat.2009.03.023

Examples

```
water_productivity(4500, 500, unit = "mm")
water_productivity(4500, 500, price = 25, unit = "mm")
```

`water_use_efficiency` *Water Use Efficiency (WUE)*

Description

Computes water use efficiency as the ratio of economic yield to total water consumed (evapotranspiration or irrigation plus rainfall).

Usage

```
water_use_efficiency(yield, water_consumed, verbose = TRUE)
```

Arguments

`yield`	Numeric vector. Crop yield (kg/ha).
`water_consumed`	Numeric vector. Total water consumed (mm), typically evapotranspiration (ET), or irrigation applied plus effective rainfall.
`verbose`	Logical. If `TRUE`, prints informational messages. Default `TRUE`.

Details

Water use efficiency is defined as:

$$WUE = \frac{Y}{W}$$

where Y is economic yield (kg/ha) and W is total water consumed (mm).

Value

Numeric vector. Water use efficiency in kg/ha/mm. Higher values indicate more efficient use of water for crop production.

References

Hoover, D.L. et al. (2023). Indicators of water use efficiency across diverse agroecosystems and scales. *Science of the Total Environment*, 864, 160992. doi:10.1016/j.scitotenv.2022.160992

Examples

```
# Wheat yield with different irrigation levels
yield <- c(4500, 4200, 3800, 3500)
water <- c(450, 400, 350, 300)
water_use_efficiency(yield, water)
```

Index