

Package: reiod (via r-universe)

October 27, 2026

Title Dietary Environmental Impact Metrics from the Intake EIOD Framework

Version 0.1.0

Description Calculates the dietary environmental impact metrics defined in Intake's Environmental Impacts of Foods methods paper: absolute threshold metrics against per capita planetary boundary shares, relative benchmark metrics against a fixed global reference distribution, and the two composite metrics that combine diet quality with environmental impact. Cut-offs are age- and sex-specific. The package does not contain the Environmental Impacts of Foods database itself; it applies the published metric cut-offs to impact values the user has already derived. Includes a data quality check suite covering variable type, plausibility, numerical precision and unit magnitude.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5)

Imports stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Language en-GB

URL <https://nicholustintzaw.github.io/reiod/>,
<https://github.com/nicholustintzaw/reiod>

BugReports <https://github.com/nicholustintzaw/reiod/issues>

Config/roxygen2/version 8.1.0

Repository <https://ropensci-staging.r-universe.dev>

Date/Publication 2026-10-27 05:09:18 UTC

RemoteUrl <https://github.com/nicholustintzaw/reiod>

RemoteRef HEAD
RemoteSha dcd76d72c582dadcd579b87149ba39cc0a80d5541

Contents

eiod_absolute_all	2
eiod_absolute_threshold	4
eiod_age_sex_group	6
eiod_as_sex_factor	8
eiod_check_data	8
eiod_check_demographics	10
eiod_check_gdqs	11
eiod_check_impact	12
eiod_check_magnitude	13
eiod_check_precision	14
eiod_cutoffs	15
eiod_demo	18
eiod_demo_map	19
eiod_hdlei	20
eiod_healthy_diet	21
eiod_hesd	22
eiod_indicators	23
eiod_list_indicators	24
eiod_lookup_cutoff	25
eiod_low_impact	27
eiod_relative_all	28
eiod_relative_band	29
eiod_sustainable_diet	31
Index	32

eiod_absolute_all	<i>All absolute threshold metrics at once (Table 3)</i>
-------------------	---

Description

Applies [eiod_absolute_threshold()] to several indicators in one call and returns a data frame of results, one column per indicator.

Usage

```
eiod_absolute_all(data, group, map, prefix = "abs_")
```

Arguments

data	A data frame with one row per respondent-day.
group	Integer vector of demographic groups 1-8, from <code>[eiod_age_sex_group()]</code> , of length <code>'nrow(data)'</code> .
map	Named character vector: names are indicator keys, values are column names in <code>'data'</code> . See <code>[eiod_demo_map]</code> for the shape.
prefix	Prefix for the returned column names. Default <code>"abs_"</code> .

Value

A data frame with `'nrow(data)'` rows and one integer column per indicator in `'map'`, named `'paste0(prefix, indicator)'`.

The map is the point

`'map'` states, explicitly, which column of your data holds which indicator. The `**name**` is the indicator key `reiod` knows (`"ghg"`); the `**value**` is whatever your column is actually called (`"ghg_kgco2e"`). They are different things, and nothing in `reiod` has ever required them to match - the vector API means `'eiod_absolute_threshold(d$anything, grp, "ghg")'` works.

Supply any subset. Indicators you do not name are simply absent from the result; there is no need to have all seven.

See Also

`[eiod_absolute_threshold()]` for the single-indicator form and the full discussion of the metric; `[eiod_relative_all()]`.

Examples

```
data(eiod_demo)
grp <- eiod_age_sex_group(eiod_demo$age, eiod_demo$sex)

# your own column names, stated once
res <- eiod_absolute_all(eiod_demo, grp,
                        map = c(ghg = "ghg_kgco2e", biodiv = "biodiv_pdf"))
colMeans(res, na.rm = TRUE)

# all seven, using the mapping shipped with the demo data
all7 <- eiod_absolute_all(eiod_demo, grp, eiod_demo_map)
round(100 * colMeans(all7, na.rm = TRUE), 1)
```

eiod_absolute_threshold

Absolute threshold metric (Table 3)

Description

Codes a one-day diet '1' if its per capita environmental impact falls below the demographic group's share of the per capita daily planetary boundary, and '0' if it does not.

Usage

```
eiod_absolute_threshold(value, group, indicator, details = FALSE)
```

Arguments

value	Numeric vector of per capita daily impact, already summed across everything the respondent consumed that day, in the unit shown by [eiod_list_indicators()].
group	Integer vector of demographic groups 1-8, from [eiod_age_sex_group()].
indicator	One of "landtotal", "landarable", "landpasture", "ghg", "eutroph", "water", "biodiv".
details	If 'TRUE', return a data frame that also reports the threshold each observation was actually compared against, rather than just the result. Default 'FALSE'.

Value

An integer vector: '1' below the threshold, '0' at or above it, 'NA' where the impact or the group is missing.

If 'details = TRUE', a data frame with one row per observation and the columns 'value', 'group', 'cutoff' and 'below'. 'cutoff' is the Table 3 threshold applied to that row, so a reader can audit the comparison without opening the reference table. (Equivalent to 'threshvar()' in the companion Stata command.) The columns are the same for every indicator; an extra 'value_kg' column existed only while eutrophication needed converting, and is gone.

Interpretation

"x% of one-day diets fall below the absolute per capita daily threshold established for \[indicator\]."

This is a pass/fail against a planetary boundary share. It says nothing about how the diet compares with other diets; for that, see [eiod_relative_band()]. The two metric families answer different questions and will not agree.

The comparison is strict

The published cut-offs carry a '<' sign, so a diet sitting exactly on the threshold is **not** below it and is coded '0'.

Missing values

A missing impact returns 'NA', not '0'. This matters more than it looks: in R, 'NA < threshold' is 'NA', but a careless implementation using 'ifelse(value < threshold, 1, 0)' on a comparison that has already been coerced can turn missing data into an apparent threshold breach. Here the guard is explicit.

No unit conversion happens here, for any indicator

Between July and October 2026 this function divided eutrophication values by 1000, because the methods paper printed the Table 3 eutrophication row in kilograms while Table 4f and the EIOF database used grams.

****Intake confirmed in October 2026 that both tables are in grams**** and supplied corrected Table 3 thresholds. They are in [eiod_cutoffs], flagged by its 'value_source' column, and every indicator now compares directly.

****Supply eutrophication in grams and convert nothing.**** If you are revisiting a script that divided by 1000 for this function, remove the division. See 'vignette("eutrophication")'.

Storage type is checked here, not only in the check suite

An integer 'value' is refused with an error. 'is.numeric(1L)' is 'TRUE', so an integer column would otherwise pass every guard and be compared against the threshold with its fractional part already truncated away. Measured on 200,000 simulated respondent-days in the companion Stata implementation, that biased mean greenhouse gas emissions down by 13%. Convert with 'as.double()'.

This is the only data-quality test on the default path. The fuller suite - sign, zeros, unit magnitude, retained significant digits, cut-off proximity - is in [eiod_check_data()] and [eiod_check_precision()], and is ****not**** run for you. Run it first; see 'vignette("precision")'.

See Also

[eiod_relative_band()], [eiod_absolute_all()], [eiod_lookup_cutoff()], [eiod_check_data()]

Examples

```
data(eiod_demo)
grp <- eiod_age_sex_group(eiod_demo$age, eiod_demo$sex)

# the column is called ghg_kgco2e; the indicator key is "ghg".
# They are different things, and your data will not use the keys.
ghg_ok <- eiod_absolute_threshold(eiod_demo$ghg_kgco2e, grp, "ghg")
table(ghg_ok, useNA = "ifany")

# which number each respondent was compared against
head(eiod_absolute_threshold(eiod_demo$ghg_kgco2e, grp, "ghg",
                             details = TRUE))

# all seven indicators: state the mapping from indicator key to column
eiod_absolute_all(eiod_demo, grp, eiod_demo_map)[1:3, ]
```

eiod_age_sex_group	<i>Build the eight-category age and sex demographic group</i>
--------------------	---

Description

Every cut-off in the Intake framework is scaled to one of eight age/sex groups, because average dietary energy requirement differs across them. This function builds that grouping variable, which all other ‘eiod_*’ metric functions require.

Usage

```
eiod_age_sex_group(
  age,
  sex,
  age_units = c("years", "months"),
  male = NULL,
  female = NULL
)
```

Arguments

age	Numeric vector of respondent ages.
sex	Factor, character or numeric vector of respondent sex. See "Which type should ‘sex’ be?" above.
age_units	Either “years” (the default) or “months”.
male, female	Optional. The value of ‘sex’ that means male and female respectively. Required when ‘sex’ is numeric; optional for factor and character input, where they override the automatic label matching. Supply both or neither.

Value

An integer vector of group codes 1 to 8, with ‘NA’ where age or sex is missing. The result carries a “labels” attribute giving the group names, and can be turned into a labelled factor with ‘factor(x, 1:8, attr(x, "labels"))’.

The eight groups

- | | |
|---|-----------------|
| 1 | Male 0-1.9 yr |
| 2 | Male 2-4.9 yr |
| 3 | Male 5-9.9 yr |
| 4 | Male 10+ yr |
| 5 | Female 0-1.9 yr |
| 6 | Female 2-4.9 yr |
| 7 | Female 5-9.9 yr |
| 8 | Female 10+ yr |

Age boundaries are half-open

The paper prints the groups as "0-1.9", "2-4.9", "5-9.9" and "10+". These are read as half-open intervals: 'age < 2', '2 <= age < 5', '5 <= age < 10', 'age >= 10'. A child aged 1.95 years therefore belongs in the first group. Testing 'age <= 1.9' instead would leave everyone between 1.9 and 2.0 unassigned, and they would then drop out of every downstream metric without any error being raised.

Which type should 'sex' be?

All three common types are accepted, but they are not equally safe.

A factor is the recommended input. Its levels carry the meaning, so the data is self-documenting and no separate codebook is needed; 'table()' and model output print "male" and "female" rather than 1 and 2; and the function can resolve it without being told anything further.

Character works identically and is what 'read.csv()' produces by default in R 4.0 and later. Converting it to a factor costs nothing and is worth doing.

Numeric requires 'male' and 'female' to be given explicitly. This is a deliberate refusal rather than a missing feature. Survey data uses 1/2 and 0/1 about equally often, and there is no way to tell them apart from the values alone: both are just two distinct numbers. If the function guessed and guessed wrong, every result would be silently inverted, with no error and nothing visibly odd in the output. Rather than carry that risk, numeric input errors until the codes are stated.

The recommended route for numeric data is to convert once, near the top of the analysis, with [eiod_as_sex_factor()]. After that every downstream call is unambiguous.

For factor and character input, these labels are recognised automatically, case-insensitively and ignoring surrounding spaces: 'm', 'male', 'males', 'man', 'men', 'boy', 'boys', and 'f', 'female', 'females', 'woman', 'women', 'girl', 'girls'. Anything else raises an error listing the unrecognised values; they are never dropped silently, because that would quietly shrink the analysis sample.

See Also

[eiod_as_sex_factor()] to convert a numeric sex code once.

Examples

```
# the recommended input: sex as a factor
eiod_age_sex_group(age = c(1.5, 3, 7, 40), sex = factor(c("male", "male",
  "female", "female")))

# character works the same way
eiod_age_sex_group(age = c(1.5, 40), sex = c("M", "F"))

# numeric requires the codes to be stated
eiod_age_sex_group(age = c(1.5, 40), sex = c(1, 2), male = 1, female = 2)

# age in months
eiod_age_sex_group(age = c(23, 24), sex = c("male", "male"),
  age_units = "months")
```

eiod_as_sex_factor	<i>Convert a sex variable to the recommended factor form</i>
--------------------	--

Description

Turns a numeric or character sex variable into a two-level factor with the levels “male” and “female”. Doing this once near the top of an analysis removes the 0/1 versus 1/2 ambiguity from every later call.

Usage

```
eiod_as_sex_factor(x, male, female, labels = c("male", "female"))
```

Arguments

x	A numeric or character vector of sex codes.
male, female	The value of ‘x’ that means male and female respectively. Both are required: see the discussion in [eiod_age_sex_group()] for why this is not inferred.
labels	The two level labels to use, male first.

Value

A factor with two levels, ‘NA’ preserved.

Examples

```
eiod_as_sex_factor(c(1, 2, 2, 1), male = 1, female = 2)
eiod_as_sex_factor(c(0, 1, 1), male = 1, female = 0)
eiod_as_sex_factor(c("M", "F"), male = "M", female = "F")
```

eiod_check_data	<i>Check a dataset before computing any metric</i>
-----------------	--

Description

Runs every ‘eiod_check_*’ function over a data frame and returns a single report. This is the function to call first, before [eiod_absolute_threshold()] or anything else.

Usage

```
eiod_check_data(
  data,
  impacts = NULL,
  age = NULL,
  sex = NULL,
  gdqs = NULL,
  age_units = c("years", "months"),
  male = NULL,
  female = NULL,
  tolerance = 1e-06
)

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

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

Arguments

<code>data</code>	A data frame.
<code>impacts</code>	A named character vector mapping indicator names to column names, for example <code>'c(biodiv = "biodiv_pdf", ghg = "ghg_kg")'</code> . Names must be among the seven in <code>[eiod_list_indicators()]</code> .
<code>age, sex, gdqs</code>	Column names, or <code>'NULL'</code> to skip that check.
<code>age_units</code>	<code>"years"</code> (default) or <code>"months"</code> .
<code>male, female</code>	Optional sex codes; see <code>[eiod_age_sex_group()]</code> .
<code>tolerance</code>	Relative distance from a cut-off below which a classification is treated as not robust. Default <code>'1e-6'</code> .
<code>x</code>	An <code>'eiod_check'</code> object.
<code>...</code>	Ignored.
<code>object</code>	An <code>'eiod_check'</code> object.

Value

An object of class `'eiod_check'`. Use `'summary()'` for a single pass/fail, or treat it as a data frame to filter the findings.

What it checks

- **Structure** - that each named column exists, and that impact columns are doubles rather than factors, characters or integers.
- **Validity** - no negative or non-finite impacts, exact zeros flagged, missingness reported.
- **Precision** - significant digits actually retained, and how many observations sit close enough to a cut-off that rounding could change their classification. See `[eiod_check_precision()]`.

- **Unit magnitude** - each impact compared against the published global reference, with likely unit errors named. See [eiod_check_magnitude()].
- **Demographics** - age range and units, sex coding, coverage of the eight groups, and the count of under-twos.
- **GDQS** - type and range.

Severity

‘error’ means a metric computed on this data would be wrong, not merely uncertain; fix it first. ‘warning’ means proceed, but you need to have looked. ‘note’ is information worth having in the record.

See Also

[eiod_check_impact()], [eiod_check_precision()], [eiod_check_magnitude()], [eiod_check_demographics()], [eiod_check_gdqs()]

Examples

```
data(eiod_demo)

report <- eiod_check_data(
  eiod_demo,
  impacts = c(biodiv = "biodiv_pdf", ghg = "ghg_kgco2e"),
  age = "age", sex = "sex", gdqs = "gdqs"
)
report
summary(report)
```

```
eiod_check_demographics
```

Check the age and sex variables

Description

Verifies that age and sex can be turned into the eight demographic groups, and reports group coverage, including how many respondents fall below the age-2 floor that the composite metrics apply.

Usage

```
eiod_check_demographics(
  age,
  sex,
  age_units = c("years", "months"),
  male = NULL,
  female = NULL
)
```

Arguments

age	Numeric vector of ages.
sex	Factor, character or numeric vector of sex.
age_units	"years" (default) or "months".
male, female	Optional codes; see [eiod_age_sex_group()].

Details

Like the other check functions, this reports rather than stops. Where [eiod_age_sex_group()] would raise an error, this records it as a finding so you can see every problem in one pass instead of one per run.

Value

An object of class 'eiod_check'.

Examples

```
data(eiod_demo)
eiod_check_demographics(eiod_demo$age, eiod_demo$sex)

# numeric sex with no codes given: reported, not guessed
eiod_check_demographics(eiod_demo$age,
  as.integer(eiod_demo$sex == "male"))
```

eiod_check_gdqs	<i>Check the Global Diet Quality Score variable</i>
-----------------	---

Description

Check the Global Diet Quality Score variable

Usage

```
eiod_check_gdqs(gdqs, cutoff = 23, max_score = 49)
```

Arguments

gdqs	Numeric vector of GDQS values.
cutoff	The healthy-diet threshold, default 23.
max_score	The maximum attainable score, used for the range check. Default 49.

Value

An object of class 'eiod_check'.

Examples

```
data(eiod_demo)
eiod_check_gdqs(eiod_demo$gdqs)
```

eiod_check_impact	<i>Check an environmental impact variable</i>
-------------------	---

Description

Structural and plausibility checks on one impact vector: storage type, sign, zeros, missingness and non-finite values.

Usage

```
eiod_check_impact(value, name = "impact", indicator = NULL)
```

Arguments

value	The impact vector to check. May be any type; that is the point.
name	A label for the variable, used in the report.
indicator	Optionally the indicator name, used to report the expected unit.

Value

An object of class 'eiod_check'.

Why the type checks are errors

Three import failures destroy impact data silently, and all three are common with the very small values biodiversity loss takes:

- A **factor** column. 'as.numeric()' on a factor returns the level codes, so '3.7e-13' becomes '1'. A plausible-looking number with no warning is far worse than 'NA', because nothing downstream looks wrong.
- A **character** column, usually caused by one stray cell such as "<0.001" or a decimal comma. Coercion then yields 'NA' with a warning that is easy to miss.
- An **integer** column. It cannot represent '1e-13' at all; the value is already '0' by the time you see it.

Examples

```
data(eiod_demo)
eiod_check_impact(eiod_demo$biodiv_pdf, name = "biodiv_pdf")

# a factor column, the worst of the three import failures
eiod_check_impact(factor(c("3.7e-13", "5.2e-13")), name = "biodiv")
```

eiod_check_magnitude	<i>Check the magnitude of an impact variable against the global reference</i>
----------------------	---

Description

Compares the median of your data against the published global 50th percentile for that indicator, weighted to your sample's demographic mix, and names the likely conversion when the ratio lands near a common unit factor.

Usage

```
eiod_check_magnitude(value, indicator, group = NULL, name = "impact")
```

Arguments

value	Numeric vector of impact values.
indicator	The indicator name.
group	Demographic groups 1-8, used to weight the expected median.
name	A label for the variable, used in the report.

Details

This is the check that catches a unit error. A greenhouse gas column supplied in grams rather than kilograms will sit about 1000 times above the reference and be reported as such, rather than quietly producing a result in which every diet exceeds every threshold.

Value

An object of class 'eiod_check'.

Examples

```
data(eiod_demo)
grp <- eiod_age_sex_group(eiod_demo$age, eiod_demo$sex)

eiod_check_magnitude(eiod_demo$ghg_kgco2e, "ghg", grp, name = "ghg_kgco2e")

# the same data in grams
eiod_check_magnitude(eiod_demo$ghg_kgco2e * 1000, "ghg", grp,
  name = "ghg_grams")
```

eiod_check_precision *Check the numerical precision of an impact variable*

Description

Two related checks: how many significant digits the stored values actually retain, and how many observations sit close enough to their own cut-off that the classification could flip under rounding.

Usage

```
eiod_check_precision(
  value,
  group = NULL,
  indicator = NULL,
  name = "impact",
  tolerance = 1e-06,
  metric = c("abs", "rel"),
  percentile = NULL
)
```

Arguments

value	Numeric vector of impact values.
group	Optional demographic groups 1-8. Required for the cut-off proximity check.
indicator	Optional indicator name. Required for the proximity check.
name	A label for the variable, used in the report.
tolerance	Relative distance from a cut-off below which a classification is treated as not robust. Default '1e-6'.
metric	Which cut-off to measure distance from: "abs" (default) or "rel".
percentile	Required when 'metric = "rel"'.

Value

An object of class 'eiod_check'.

The small exponent is not the problem

R's 'numeric' is an IEEE 754 double: around 15 to 17 significant decimal digits, with an exponent range to about $1e\pm308$. A biodiversity value of '3.6896e-13' is comfortably representable, and comparison with '<' is exact. No epsilon is needed, provided both sides carry full precision.

The real risk is **ingestion fidelity** - digits lost before R ever sees the number. Excel displays '3.69E-13' and can write back only what it displays; CSV exports written with a 'float' storage holds about 7 significant digits; database 'REAL' columns are single precision for the same reason.

What the checks measure

The first finds, for each distinct value, the smallest number of significant digits that reproduces it exactly. Across many distinct values, a maximum of 7 or fewer is consistent with single-precision storage or a truncated export. Measured on simulated data, full-precision values return 17, a single-precision round trip returns exactly 7, and an Excel three-digit re-save returns 3, so the cases separate cleanly.

The second is the one that answers whether precision matters *here*. It computes, for each observation, the relative distance to the cut-off that applies to it, and counts those within ‘tolerance’. Usually the answer is zero and the worry can be set aside; when it is not, you get the row numbers.

Nothing is repaired. Digits that were never read cannot be recovered.

Examples

```
data(eiod_demo)
grp <- eiod_age_sex_group(eiod_demo$age, eiod_demo$sex)
eiod_check_precision(eiod_demo$biodiv_pdf, grp, "biodiv",
  name = "biodiv_pdf")

# the same values after a single-precision round trip
eiod_check_precision(signif(eiod_demo$biodiv_pdf, 7), grp, "biodiv",
  name = "biodiv_float")
```

eiod_cutoffs

Published cut-offs for every dietary environmental impact metric

Description

Every cut-off in Tables 3 and 4a-4g of the Intake methods paper, in long format: one row per metric, indicator, demographic group and table row.

Usage

```
eiod_cutoffs
```

Format

A data frame with 616 rows and 15 columns:

metric “abs” for Table 3, “rel” for Tables 4a-4g

table Source table: “3”, “4a” ... “4g”

indicator Short indicator name used by the ‘eiod_*’ functions

indicator_label Human-readable indicator name

unit Unit *this table is printed in*, per capita per day. The same for both tables for six indicators; for eutrophication, kg for Table 3 and g for Table 4f

demogroup Demographic group, 1-8

demogroup_label Group name, e.g. "Male 10+ yr"

sex Factor, "male" or "female"

age_min, age_max Age bounds in years; 999 means no upper bound

row_order Position in the source table

percentile 10-90 for relative rows 1-9; 'NA' for absolute rows and for row 10, which repeats P90 as a lower bound

value ****The cut-off the metric functions compare against.**** Nothing is rescaled: for 608 rows this is exactly what the paper prints, and for the other eight it is exactly what Intake supplied

value_published_str The same value as a character string, exactly as received. Shipped so the figure can be compared character for character, not as a rendering of a double

value_source "published" for the 608 rows transcribed from the paper; "intake correction 2026-10" for the eight Table 3 eutrophication thresholds Intake corrected. ****Those eight do not match the currently published PDF**** - see the section above

Details

Tables 5a and 5b are deliberately ****not**** stored here. They are composites that reuse values already present - the HESD criterion is 'metric == "abs" & indicator == "biodiv"', and the HDLEI criterion is 'metric == "rel" & indicator == "biodiv" & percentile == 30'. A second copy could drift from the first.

Nothing here is rescaled

****'value' is the cut-off exactly as this package received it, for all 616 rows****, and 'value_published_str' is the same number as literal text. The package applies no conversion to a cut-off and none to your data.

For 608 rows "as received" means "as printed in the methods paper", and you can check them against the PDF character for character. The remaining eight are described in the next section.

Eight rows differ from the published PDF, on Intake's instruction

The ****Table 3 eutrophication thresholds**** in 'eiod_cutoffs' are ****not**** the numbers the currently published paper prints. They are corrections supplied by Intake in October 2026, and the 'value_source' column marks them:

```
subset(eiod_cutoffs, value_source != "published",
       c(demogroup_label, unit, value, value_source))
```

****What was wrong.**** The paper prints those eight cells in ****kilograms**** PO4(3-)eq while Table 4f, and the EIOF food databases, use ****grams****. Intake confirmed by email that both tables are intended to be in grams, and supplied the corrected Table 3 values.

****What changed, and what did not.**** The corrected values are the printed ones multiplied by 1000, carried to full precision - the quantity is identical, only the unit it is expressed in has changed. Dividing any corrected value by 1000 reproduces the printed figure to the ten decimals the PDF gives, and 'data-row/build_data.R' asserts exactly that. ****No classification moves****: tested on 400,000 simulated gram-scale values, old and new behaviour agreed everywhere except within

about one unit in the last place of a cut-off, where the new path is the more exact of the two because it no longer divides before comparing.

****Consequence for you: supply eutrophication in grams**, which is what the EIOF databases give you ('Eutr. (g PO43-e/kg)' x kg food = g). Every indicator now uses one unit across both tables, and nothing is converted anywhere. Between July and October 2026 this package divided eutrophication values by 1000 on your behalf; that step is gone.**

See 'vignette("eutrophication")' for the full history, including the evidence that led to the question being raised.

Checking a value against the printed table

Two columns carry the same number in two forms, so that a reader can check the package without having to trust any numeric rendering:

'value_published_str'	the literal characters from the paper, never through a numeric type
'value'	the nearest IEEE 754 double to that text - and what the metric functions compare against

'data-raw/build_data.R' asserts that 'as.numeric(value_published_str) == value' for all 616 rows, and that each parsed value still prints back to exactly its published digits.

A 'double' cannot hold most decimal fractions exactly - that is true of every statistical package, not of this one. What is guaranteed here is that the stored double is the nearest double to the published decimal, that it round-trips back to that decimal, and that the decimal itself is in the file as text. ****If you need an exact decimal comparison, compare against 'value_published_str'.**

Note that R's default 'options(digits = 7)' will not show you a published value in full: 15 of the 616 need 14 significant digits. [eiod_lookup_cutoff()] prints at full precision for this reason.

Source

Intake-EIOD-Methods-Paper-Jan2026-Final.pdf, Tables 3 and 4a-4g (pages 16-19), available at <<https://www.intake.org/sites/default/files/2026-01/Intake-EIOD-Methods-Paper-Jan2026-Final.pdf>>. Transcribed programmatically; see 'data-raw/build_data.R'. The same source file drives a companion Stata implementation.

Examples

```
data(eiod_cutoffs)
str(eiod_cutoffs)

# the HESD biodiversity criterion
subset(eiod_cutoffs, metric == "abs" & indicator == "biodiv",
       c(demogroup_label, value))
```

eiod_demo

*Synthetic respondent-day data for examples***Description**

Six hundred simulated respondent-days with age, sex, the seven impact totals and a GDQS score, for use in examples, vignettes and tests.

Usage

eiod_demo

Format

A data frame with 600 rows and 11 columns:

id Row identifier

age Age in years, 0.5 to 65

sex Factor with levels "male" and "female"

land_total_m2yr, land_arable_m2yr, land_pasture_m2yr Land use, m2*year

ghg_kgco2e Greenhouse gas emissions, kg CO2eq

eutroph_gpo4e Eutrophication potential, g PO4(3-)eq. Grams, like the EIOF databases and both cut-off tables. Renamed from eutroph_kgpo4e and rescaled in 0.1.0; see vignette("eutrophication")

water_litres Water use, litres

biodiv_pdf Biodiversity loss, PDF*year

gdqs Global Diet Quality Score

Details

****These are not real survey data.**** The values were drawn from lognormal distributions chosen to be roughly plausible. No prevalence computed from this dataset carries any substantive meaning.

A small amount of missingness is included in 'biodiv_pdf' and 'gdqs' so examples show how it propagates.

The column names are deliberately not the indicator keys

Until October 2026 the impact columns were called 'landtotal', 'ghg', 'biodiv' and so on - the same strings the 'indicator' argument takes. That made the examples tidy and taught something false: a loop over 'eiod_demo[[ind]]' works only because the columns happen to be named after the keys, and readers concluded that reiod required those names.

It never did. reiod takes vectors, so any column name has always worked. The columns now carry survey-realistic names with their units, which forces every example to state the indicator-to-column mapping explicitly. See [eiod_demo_map] and [eiod_absolute_all()].

See Also

[eiod_demo_map] for the indicator-to-column mapping. A deliberately broken copy ships as ‘system.file("extdata", "eiod_broken.csv", package = "reiod")’ and is used in ‘vignette("troubleshooting”)’.

Examples

```
data(eiod_demo)
head(eiod_demo)
```

eiod_demo_map	<i>Indicator keys mapped to the demonstration data’s column names</i>
---------------	---

Description

A named character vector in the shape every ‘*_all()’ function expects: the **name** is the indicator key reiod knows, the **value** is the column in [eiod_demo] that holds it.

Usage

```
eiod_demo_map
```

Format

A named character vector of length 7.

Details

It is shipped rather than retyped in each example, and it exists at all because [eiod_demo]’s columns are deliberately *not* named after the indicator keys. Write the equivalent line for your own data once, near the top of your script, and every later call reads from it.

See Also

[eiod_absolute_all()], [eiod_relative_all()], [eiod_demo]

Examples

```
data(eiod_demo_map)
eiod_demo_map

# the same shape, for your own data:
# my_map <- c(ghg = "ghg_total_kg", biodiv = "pdf_year", water = "water_1")
```

eiod_hdlei

*Healthy Diet with Low Environmental Impacts, HDLEI (Table 5b)***Description**

Combines the same diet quality criterion as [eiod_hesd()] with a **relative** low-impact criterion.

Usage

```
eiod_hdlei(
  biodiversity,
  gdqs,
  group,
  form = c("category", "binary", "components"),
  gdqs_cutoff = 23,
  percentile = 30
)
```

Arguments

biodiversity	Numeric vector of per capita daily biodiversity loss.
gdqs	Numeric vector of Global Diet Quality Score values.
group	Integer vector of demographic groups 1-8.
form	"category", "binary" or "components"; see [eiod_hesd()].
gdqs_cutoff	GDQS threshold, default 23.
percentile	Relative cut-off, default 30 as published.

Details

- healthy: GDQS ≥ 23
- low environmental impacts: biodiversity loss below the 30th percentile of the Table 4g relative benchmark distribution

The diet quality criterion is identical to HESD's. The only difference is the environmental one: HESD tests biodiversity against the absolute planetary boundary share, HDLEI against a position in the global reference distribution.

Value

A factor, integer vector, or data frame, depending on 'form'.

These two metrics are nested

The HDLEI environmental cut-off is uniformly about 1.52 times stricter than the HESD one, in every demographic group, because both are the same all-population figure scaled by the same weight. So 'eiod_hdlei(...) == 1' implies 'eiod_hesd(...) == 1', and HESD prevalence can never fall below HDLEI prevalence. Reporting both is still informative: the gap between them is the share of diets sitting between the two cut-offs. But they are not two independent dimensions.

Age restriction

As for HESD, results are 'NA' for respondents under 2 years.

Examples

```
data(eiod_demo)
grp <- eiod_age_sex_group(eiod_demo$age, eiod_demo$sex)

table(eiod_hdlei(eiod_demo$biodiv_pdf, eiod_demo$gdqs, grp))

# the nesting: every HDLEI diet is also a HESD diet
a <- eiod_hesd(eiod_demo$biodiv_pdf, eiod_demo$gdqs, grp, form = "binary")
b <- eiod_hdlei(eiod_demo$biodiv_pdf, eiod_demo$gdqs, grp, form = "binary")
table(HESD = a, HDLEI = b, useNA = "ifany")
```

eiod_healthy_diet	<i>Healthy diet criterion</i>
-------------------	-------------------------------

Description

The diet quality half of both composite metrics: a Global Diet Quality Score of 23 or more. Exported on its own so it can be reported separately from the composites.

Usage

```
eiod_healthy_diet(gdqs, cutoff = 23)
```

Arguments

gdqs	Numeric vector of Global Diet Quality Score values.
cutoff	The score at or above which a diet counts as healthy. Default 23, as published. Change only with a documented reason.

Value

An integer vector: '1' healthy, '0' not, 'NA' where 'gdqs' is missing.

Examples

```
eiod_healthy_diet(c(10, 22.9, 23, 30, NA))
```

eiod_hesd

*Healthy and Environmentally Sustainable Diet, HESD (Table 5a)***Description**

Combines a diet quality criterion with an **absolute** environmental sustainability criterion.

Usage

```
eiod_hesd(
  biodiversity,
  gdqs,
  group,
  form = c("category", "binary", "components"),
  gdqs_cutoff = 23
)
```

Arguments

biodiversity	Numeric vector of per capita daily biodiversity loss.
gdqs	Numeric vector of Global Diet Quality Score values.
group	Integer vector of demographic groups 1-8.
form	What to return. "category" (default) gives the four categories of Table 5a as a factor; "binary" gives 1 for category 1 and 0 otherwise; "components" gives a data frame with both sub-indicators and both combined forms.
gdqs_cutoff	GDQS threshold, default 23.

Details

- healthy: GDQS ≥ 23
- environmentally sustainable: biodiversity loss below the Table 3 absolute threshold

Both criteria are also available on their own, as `eiod_healthy_diet()` and `eiod_sustainable_diet()`. This function calls those two rather than re-implementing them, so the sub-indicators and the composite cannot drift apart.

Value

A factor, integer vector, or data frame, depending on 'form'.

Age restriction

The methods paper restricts the composite metrics to respondents aged 2 years and older, because the GDQS has not been validated below that age. Table 5a prints six demographic columns, not eight. Results are therefore 'NA' for groups 1 and 5, and a message reports how many observations that affects. If your sample includes under-tuos, decide explicitly whether they belong in the analysis denominator at all.

Relationship to HDLEI

HESD and [eiod_hdlei()] are nested, not independent: every HDLEI diet is also a HESD diet. See [eiod_low_impact()] for why. Do not present them as two dimensions of a 2x2.

Examples

```
data(eiod_demo)
grp <- eiod_age_sex_group(eiod_demo$age, eiod_demo$sex)

table(eiod_hesd(eiod_demo$biodiv_pdf, eiod_demo$gdqs, grp))

# everything at once
parts <- eiod_hesd(eiod_demo$biodiv_pdf, eiod_demo$gdqs, grp,
                  form = "components")
head(parts)
```

eiod_indicators	<i>The seven environmental impact indicators</i>
-----------------	--

Description

Indicator names accepted by the ‘eiod_*’ functions, with their labels and the units your impact values must be in.

Usage

```
eiod_indicators
```

Format

A data frame with 7 rows and 3 columns:

indicator Short name, as passed to the ‘eiod_*’ functions

indicator_label Human-readable name

unit Required unit, per capita per day

Examples

```
data(eiod_indicators)
eiod_indicators
```

eiod_list_indicators *List the seven environmental impact indicators*

Description

Prints the indicator names the 'eiod_*' functions accept, with their labels and the units your impact values must be in.

Usage

```
eiod_list_indicators(quiet = FALSE)
```

Arguments

quiet If 'TRUE', return the table without printing it.

Value

A data frame of indicator, label and unit, invisibly when printing.

If your data comes from the EIOF database

****Supply it as it comes - nothing needs converting.**** The units in this table are the EIOF database's own: 'Land (m2/kg)', 'GHG (kg CO2e/kg)', 'Eutr. (g PO43-e/kg)', 'Water Use (L/kg)'. Multiply by kilograms of food consumed, sum to the respondent-day, and pass the result straight in.

kg food consumed x EIOF intensity per kg = the value this package wants

****This changed in October 2026.**** Eutrophication used to be the exception: the methods paper printed its Table 3 threshold in kilograms while the database reported grams, so a conversion was needed and this package applied one. Intake has since confirmed that both cut-off tables are in grams and supplied corrected Table 3 values. Every indicator is now on a single unit and no conversion happens anywhere.

If you are revisiting a script written before that change and it divides eutrophication by 1000, ****remove the division**** - the package no longer expects it. 'eiod_cutoffs' has the detail.

Values must be summed over ALL consumed foods - including any the EIOF database cannot cost

Not every food has an impact factor. The EIOF country databases mark foods with no available estimate, and such a food contributes ****nothing**** to the day's total - so a respondent who ate one has an impact that is understated by an unknown amount.

****This package cannot detect that.**** It receives one number per respondent-day; whether that number was summed over complete or incomplete coverage is not recoverable from the number itself. The only exception is the extreme case: if every food lacked a factor and the sum was coded '0' rather than 'NA', [eiod_check_impact()] flags the zero.

It matters more here than in most analyses because these metrics compare each respondent against a **fixed external cut-off** rather than against your sample, so uneven coverage shifts results with nothing in the output looking wrong.

Count it where it is still visible - when you aggregate to the respondent-day - preferably as the **share of consumed weight** carrying a factor rather than a count of items, and report it with any prevalence. See `vignette("troubleshooting")`.

Examples

```
eiod_list_indicators()
```

eiod_lookup_cutoff	<i>Look up a published cut-off value</i>
--------------------	--

Description

Returns the cut-off itself, rather than applying it. Useful for reporting which number a respondent was compared against, and for checking the package against the printed tables.

Usage

```
eiod_lookup_cutoff(
  indicator,
  group = NULL,
  metric = c("abs", "rel"),
  percentile = NULL,
  published = FALSE
)

## S3 method for class 'eiod_cutoff'
print(x, digits = 15, ...)

## S3 method for class 'eiod_cutoff'
Ops(e1, e2)
```

Arguments

indicator	One of the seven indicator names.
group	Demographic group 1-8, or 'NULL' for all eight.
metric	"abs" for the Table 3 absolute threshold, or "rel" for a Table 4 relative benchmark cut-point.
percentile	Required when 'metric = "rel"': a multiple of 10 from 10 to 90.

published	**No longer has any effect** , and retained only so existing scripts keep working. Until October 2026 the package stored 80 relative eutrophication rows pre-rescaled, and this argument switched between the stored and the published figure. Nothing is rescaled any more - ‘value’ is the published figure for all 616 rows - so both settings return the same thing. See ‘vignette("eutrophication")’.
x	An ‘eiod_cutoff’ object.
digits	Significant digits to display. The default of 15 shows every published digit; the widest cut-off in the tables needs 14.
...	Ignored.
e1, e2	Operands.

Value

A named numeric vector of class ‘eiod_cutoff’, named by demographic group label. Use ‘print(x, digits =)’ to change the displayed precision, or ‘as.numeric(x)’ to drop the class.

Printing, and why it has its own method

The published cut-offs need up to ****14 significant digits**** - ‘water’ for Female 10+ is ‘1726.3821057877’, and 15 of the 616 values need all 14. R’s default ‘options(digits = 7)’ would print that as ‘1726.382’, which looks as though the package had rounded the guideline.

Storage was never the problem; the default **display** was. The returned object therefore carries a class and a print method that uses 15 significant digits and shows, alongside each value, the literal text transcribed from the paper:

```
““ Male 10+ yr 14.2377570161 as printed 14.2377570161 ““
```

The ‘as printed’ column is ‘value_published_str’ from [eiod_cutoffs] - a character string that has never been through a numeric type. If you need an exact decimal comparison against the PDF, compare against that, not against a rendering of a double. (The identical complaint, and the identical fix, arose in the companion Stata package in October 2026.)

The object is an ordinary named numeric underneath: ‘[, ‘c()’ and arithmetic all return plain numbers, so nothing downstream needs to know about the class.

Nothing here is rescaled

****Every value this function returns is exactly what the methods paper prints****, for all 616 cut-offs. The package does not adjust a published number anywhere, so you can check any row of [eiod_cutoffs] against the PDF and expect a character-for-character match.

Where the paper’s own units need reconciling - which happens for eutrophication, and only for eutrophication - the adjustment is applied to ****your data**** at the moment of comparison, is announced in a message, and is described in ‘vignette("eutrophication")’.

Examples

```
# the HESD biodiversity criterion, all groups
eiod_lookup_cutoff("biodiv", metric = "abs")

# the HDLEI biodiversity criterion
```

```
eiod_lookup_cutoff("biodiv", metric = "rel", percentile = 30)

# one group, printed to every published digit
eiod_lookup_cutoff("water", group = 8, metric = "abs")

# what R's default options(digits = 7) would have shown instead
print(c(eiod_lookup_cutoff("water", group = 8, metric = "abs")))
```

eiod_low_impact

Low environmental impact criterion

Description

The environmental half of the HDLEI metric: biodiversity loss below the 30th percentile of the Table 4g *relative* benchmark distribution. Exported on its own so it can be reported separately from the composite.

Usage

```
eiod_low_impact(biodiversity, group, percentile = 30)
```

Arguments

biodiversity	Numeric vector of per capita daily biodiversity loss.
group	Integer vector of demographic groups 1-8.
percentile	The relative cut-off. Default 30, as published. Any other value is a sensitivity analysis, not the HDLEI metric, and is labelled as such by a message.

Value

An integer vector: '1' low impact, '0' not, 'NA' where missing.

Relationship to the HESD criterion

This criterion is uniformly stricter than [eiod_sustainable_diet()]. For biodiversity the relative P30 cut-off sits below the absolute threshold in every one of the eight demographic groups, by the same constant factor of about 1.52, because both are the same all-population figure scaled by the same energy-requirement weight. So a diet meeting this criterion necessarily meets the HESD one, and HESD prevalence can never fall below HDLEI prevalence. The two are nested, not independent.

Examples

```
data(eiod_demo)
grp <- eiod_age_sex_group(eiod_demo$age, eiod_demo$sex)
table(eiod_low_impact(eiod_demo$biodiv_pdf, grp), useNA = "ifany")
```

eiod_relative_all	<i>All relative benchmark metrics at once (Tables 4a-4g)</i>
-------------------	--

Description

Applies [eiod_relative_band()] to several indicators in one call and returns a data frame of results, one column per indicator.

Usage

```
eiod_relative_all(data, group, map, midpoint = FALSE, prefix = "band_")
```

Arguments

data	A data frame with one row per respondent-day.
group	Integer vector of demographic groups 1-8, of length 'nrow(data)'.
map	Named character vector: names are indicator keys, values are column names in 'data'.
midpoint	If 'TRUE', return band midpoint percentiles rather than band numbers. See [eiod_relative_band()] for why a mean band is not a mean impact.
prefix	Prefix for the returned column names. Default "band_".

Value

A data frame with 'nrow(data)' rows and one integer column per indicator in 'map'.

The map is the point

'map' states, explicitly, which column of your data holds which indicator. The ****name**** is the indicator key reiod knows ("ghg"); the ****value**** is whatever your column is actually called ("ghg_kgco2e"). They are different things, and nothing in reiod has ever required them to match - the vector API means 'eiod_absolute_threshold(d\$anything, grp, "ghg")' works.

Supply any subset. Indicators you do not name are simply absent from the result; there is no need to have all seven.

See Also

[eiod_relative_band()], [eiod_absolute_all()]

Examples

```
data(eiod_demo)
grp <- eiod_age_sex_group(eiod_demo$age, eiod_demo$sex)

bands <- eiod_relative_all(eiod_demo, grp, eiod_demo_map)
sapply(bands, function(b) round(mean(b, na.rm = TRUE), 2))
```

eiod_relative_band	<i>Relative benchmark metric (Tables 4a-4g)</i>
--------------------	---

Description

Places a one-day diet in one of ten decile bands of a global reference distribution of per capita daily dietary environmental impacts.

Usage

```
eiod_relative_band(value, group, indicator, midpoint = FALSE, details = FALSE)
```

Arguments

value	Numeric vector of per capita daily impact.
group	Integer vector of demographic groups 1-8, from [eiod_age_sex_group()].
indicator	One of the seven indicator names; see [eiod_list_indicators()].
midpoint	If 'TRUE', return the band midpoint percentile (5, 15, ... 95) instead of the band number. Default 'FALSE'.
details	If 'TRUE', return a data frame that also reports the band boundaries each observation was placed between. Default 'FALSE'.

Value

An integer vector of bands 1-10, or of midpoints if 'midpoint = TRUE'. 'NA' where the impact or the group is missing.

If 'details = TRUE', a data frame with one row per observation and the columns 'value', 'group', 'band', 'cut_lower' and 'cut_upper'. 'cut_lower' is the highest published percentile the value cleared, and is 'NA' in band 1 - correctly, because nothing is published below P10. 'cut_upper' is 'NA' in band 10 for the same reason. (Equivalent to 'cutvar()' in the companion Stata command.)

What the comparison is against

This does **not** rank respondents against each other in your survey. Each diet is compared against a fixed external reference distribution that is the same for every analysis anywhere in the world. Intake built it by taking per-country average dietary impact per person per day at each decile from the Planet-Based Diets Database, population-weighting to a global percentile distribution using 2010 country populations, and scaling the result to each demographic group by its share of average energy requirements.

The practical consequence: **do not** compute this with 'quantile()', 'cut()' or 'ntile()'. Those rank your sample against itself and will place roughly 10 different statistic with no cross-country meaning. The cut-points come from the published tables, not from your data.

If your whole sample has low impacts, your whole sample lands in low bands. That is intended, and is the reason the metric is comparable across countries and over time.

A low band is not evidence of sustainability

The reference distribution describes diets as they are, not as they would need to be. For greenhouse gas emissions the Table 3 absolute planetary-boundary threshold lies *below* the 10th percentile of the reference distribution, so a diet in band 1 still exceeds the planetary boundary. For water use the threshold sits *above* the 90th percentile, so essentially every diet passes. Report the two metric families together and never one as a proxy for the other.

Band definitions

- 1 impact < P10
- 2 P10 <= impact < P20
- ...
- 9 P80 <= impact < P90
- 10 impact >= P90

The inequality is strict, so a diet exactly on P10 belongs in band 2.

Reporting a mean band

The paper also reports a mean position. Use ‘midpoint = TRUE’ to get band midpoints (5, 15, ... 95) for that purpose, but note the limitation: averaging an ordinal band index treats the bands as equally spaced on the impact scale, and they are not. In Table 4c the P10-P20 gap is 0.002 m²*year while the P80-P90 gap is 1.5 m²*year. A mean band is a summary of *position in the reference distribution*; it is not a summary of impact. For a mean impact, average the impact variable itself.

Storage type is checked here, not only in the check suite

As for [eiod_absolute_threshold()], an integer ‘value’ is refused. See that function for why ‘is.numeric()’ alone is not enough. The rest of the data quality suite remains opt-in; run ‘vignette("precision")’ first.

See Also

[eiod_absolute_threshold()], [eiod_relative_all()], [eiod_low_impact()]

Examples

```
data(eiod_demo)
grp <- eiod_age_sex_group(eiod_demo$age, eiod_demo$sex)

# the column is biodiv_pdf; the indicator key is "biodiv"
band <- eiod_relative_band(eiod_demo$biodiv_pdf, grp, "biodiv")
table(band, useNA = "ifany")

# mean position in the global reference distribution
mid <- eiod_relative_band(eiod_demo$biodiv_pdf, grp, "biodiv",
                          midpoint = TRUE)
mean(mid, na.rm = TRUE)
```

```
# the decile boundaries each diet was judged against
head(eiod_relative_band(eiod_demo$biodiv_pdf, grp, "biodiv",
                        details = TRUE))
```

eiod_sustainable_diet *Environmentally sustainable diet criterion*

Description

The environmental half of the HESD metric: biodiversity loss below the Table 3 **absolute** threshold. Exported on its own so it can be reported separately from the composite.

Usage

```
eiod_sustainable_diet(biodiversity, group)
```

Arguments

biodiversity	Numeric vector of per capita daily biodiversity loss (PDF*year).
group	Integer vector of demographic groups 1-8.

Details

Biodiversity loss is used as the proxy for total environmental impact, following the methods paper.

Value

An integer vector: '1' sustainable, '0' not, 'NA' where missing.

See Also

[eiod_low_impact()], the stricter relative criterion used by HDLEI.

Examples

```
data(eiod_demo)
grp <- eiod_age_sex_group(eiod_demo$age, eiod_demo$sex)
table(eiod_sustainable_diet(eiod_demo$biodiv_pdf, grp), useNA = "ifany")
```

Index

* datasets

- [eiod_cutoffs](#), [15](#)
 - [eiod_demo](#), [18](#)
 - [eiod_demo_map](#), [19](#)
 - [eiod_indicators](#), [23](#)
- [eiod_absolute_all](#), [2](#)
- [eiod_absolute_threshold](#), [4](#)
- [eiod_age_sex_group](#), [6](#)
- [eiod_as_sex_factor](#), [8](#)
- [eiod_check_data](#), [8](#)
- [eiod_check_demographics](#), [10](#)
- [eiod_check_gdqs](#), [11](#)
- [eiod_check_impact](#), [12](#)
- [eiod_check_magnitude](#), [13](#)
- [eiod_check_precision](#), [14](#)
- [eiod_cutoffs](#), [15](#)
- [eiod_demo](#), [18](#)
- [eiod_demo_map](#), [19](#)
- [eiod_hdlei](#), [20](#)
- [eiod_healthy_diet](#), [21](#)
- [eiod_hesd](#), [22](#)
- [eiod_indicators](#), [23](#)
- [eiod_list_indicators](#), [24](#)
- [eiod_lookup_cutoff](#), [25](#)
- [eiod_low_impact](#), [27](#)
- [eiod_relative_all](#), [28](#)
- [eiod_relative_band](#), [29](#)
- [eiod_sustainable_diet](#), [31](#)

`Ops.eiod_cutoff (eiod_lookup_cutoff)`, [25](#)

`print.eiod_check (eiod_check_data)`, [8](#)

`print.eiod_cutoff (eiod_lookup_cutoff)`, [25](#)

`summary.eiod_check (eiod_check_data)`, [8](#)