

Package: dceasimR (via r-universe)

May 24, 2026

Title Distributional Cost-Effectiveness Analysis for Health Technology Assessment

Version 0.1.0

Description Implements distributional cost-effectiveness analysis (DCEA) as described in Cookson et al. (2020, ISBN:9780198838197) and the methods endorsed by NICE (2025) for health technology evaluation. Provides functions for both aggregate and full-form DCEA, inequality measurement (Atkinson index, Gini coefficient, slope index of inequality, relative index of inequality), social welfare function evaluation, equity-efficiency impact plane visualisation, and sensitivity analysis over inequality aversion parameters. Includes baseline health distributions for England (by IMD quintile), Canada (income quintile), and global WHO regions. Suitable for academic research, health technology assessment submissions, and public health policy analysis.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Language en-GB

URL <https://heorlytics.github.io/dceasimR/>

BugReports <https://github.com/heorlytics/dceasimR/issues>

Imports ggplot2 (>= 3.4.0), dplyr (>= 1.1.0), rlang (>= 1.1.0), tibble (>= 3.2.0), scales (>= 1.3.0), stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0), openxlsx, flextable, covr, vdiff

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 3.5)

Repository <https://heorlytics.r-universe.dev>

Date/Publication 2026-04-16 16:50:42 UTC

RemoteUrl <https://github.com/heorlytics/dceasimr>

RemoteRef HEAD

RemoteSha 5d719151467e418d957a99e11a8b20a6fde28ff8

Contents

dceasimR-package	3
build_staircase_data	4
calc_all_inequality_indices	5
calc_atkinson_index	6
calc_concentration_index	7
calc_edc	8
calc_edc_profile	9
calc_equity_weight_profile	10
calc_equity_weighted_nhb	10
calc_equity_weights	11
calc_gini	12
calc_rii	13
calc_sii	13
calc_social_welfare	14
canada_income_hale	15
compute_generalised_lorenz_data	16
compute_lorenz_data	16
compute_ridit_scores	17
england_imd_hale	18
england_imd_qol	19
example_cea_output	19
export_dcea_excel	20
generate_dcea_report	21
generate_nice_table	22
get_baseline_health	22
merge_cea_with_baseline	23
normalise_weights	24
nsclc_dcea_example	25
plot.aggregate_dcea	25
plot.full_dcea	26
plot_dcea_tornado	26
plot_edc_profile	27
plot_equity_impact_plane	28
plot_inequality_staircase	29
plot_lorenz_curve	30
prepare_subgroup_cea	31
print.aggregate_dcea	32
print.full_dcea	32
run_aggregate_dcea	33
run_dcea_sensitivity	35

run_full_dcea	36
summary.aggregate_dcea	37
summary.full_dcea	38
validate_dcea_data	38
who_regions_hale	39

Index	40
--------------	-----------

dceasimR-package	<i>dceasimR: Distributional Cost-Effectiveness Analysis for Health Technology Assessment</i>
------------------	--

Description

Implements distributional cost-effectiveness analysis (DCEA) as described in Cookson et al. (2020) and methods endorsed by NICE (2025). Provides functions for aggregate and full-form DCEA, inequality measurement, social welfare function evaluation, equity-efficiency impact plane visualisation, and sensitivity analysis.

Main functions

- [run_aggregate_dcea](#) — Aggregate DCEA (Love-Koh 2019)
- [run_full_dcea](#) — Full-form DCEA with subgroup data
- [calc_sii](#) / [calc_rii](#) — Inequality indices
- [calc_edc](#) — Equally Distributed Equivalent health
- [plot_equity_impact_plane](#) — Equity-efficiency impact plane
- [get_baseline_health](#) — Preloaded baseline distributions #'

Key references

Cookson R, Griffin S, Norheim OF, Culyer AJ (2020). *Distributional Cost-Effectiveness Analysis*. Oxford University Press. Oxford University Press (ISBN:9780198838197).

Love-Koh J, Asaria M, Cookson R, Griffin S (2019). The Social Distribution of Health: Estimating Quality-Adjusted Life Expectancy in England. *Value in Health* 22(5): 518-526. doi:10.1016/j.jval.2018.10.007

Asaria M, Griffin S, Cookson R (2016). *Distributional Cost-Effectiveness Analysis: A Tutorial*. *Medical Decision Making* 36(1): 8-19. doi:10.1177/0272989X15583266

Author(s)

Maintainer: Shubhram Pandey <shubhram.pandey@heorlytics.com> ([ORCID](#))

See Also

Useful links:

- <https://heorlytics.github.io/dceasimR/>
- Report bugs at <https://github.com/heorlytics/dceasimR/issues>

build_staircase_data *Build inequality staircase data from component inputs*

Description

Constructs the five-step inequality staircase data frame from individual component vectors. The staircase traces how the distribution of health gains is shaped at each stage: prevalence, eligibility, uptake, clinical effect, and net opportunity cost.

Usage

```
build_staircase_data(
  group,
  group_labels,
  prevalence,
  eligibility,
  uptake,
  clinical_effect,
  opportunity_cost
)
```

Arguments

group	Integer vector of group identifiers (1 = most deprived).
group_labels	Character vector of group labels.
prevalence	Numeric vector: disease prevalence by group (proportion).
eligibility	Numeric vector: proportion eligible for the intervention by group.
uptake	Numeric vector: uptake rate by group (0-1).
clinical_effect	Numeric vector: incremental QALY gain by group.
opportunity_cost	Numeric vector: QALYs displaced per group via budget impact.

Value

A tibble in long format suitable for [plot_inequality_staircase](#).

Examples

```
build_staircase_data(
  group = 1:5,
  group_labels = paste("IMD Q", 1:5),
  prevalence = c(0.08, 0.07, 0.06, 0.05, 0.04),
  eligibility = c(0.70, 0.72, 0.74, 0.76, 0.78),
  uptake = c(0.60, 0.64, 0.68, 0.72, 0.76),
  clinical_effect = c(0.30, 0.38, 0.45, 0.52, 0.58),
  opportunity_cost = c(0.05, 0.05, 0.05, 0.05, 0.05)
)
```

`calc_all_inequality_indices`*Calculate all inequality indices in one call*

Description

Convenience wrapper that computes SII, RII, concentration index, Atkinson index (for multiple ε values), and Gini coefficient and returns them as a tidy data frame.

Usage

```
calc_all_inequality_indices(  
  data,  
  health_var,  
  group_var,  
  weight_var,  
  epsilon_values = c(0.5, 1, 2)  
)
```

Arguments

<code>data</code>	A data frame with health and group columns.
<code>health_var</code>	Name of the health variable column (character).
<code>group_var</code>	Name of the socioeconomic group column (ordered integer, 1 = most deprived).
<code>weight_var</code>	Name of the population share column (sums to 1).
<code>epsilon_values</code>	Numeric vector of ε values for the Atkinson index (default: <code>c(0.5, 1, 2)</code>).

Value

A tibble with columns index, value, and description.

Examples

```
df <- tibble::tibble(  
  group      = 1:5,  
  mean_hale  = c(60, 63, 66, 69, 72),  
  pop_share  = rep(0.2, 5)  
)  
calc_all_inequality_indices(df, "mean_hale", "group", "pop_share")
```

calc_atkinson_index *Calculate Atkinson Index of Health Inequality*

Description

The Atkinson index measures the extent of inequality in the health distribution, explicitly incorporating a parameter ε representing inequality aversion. Higher ε values give more weight to health differences at the bottom of the distribution.

Usage

```
calc_atkinson_index(health_dist, pop_weights, epsilon = 1)
```

Arguments

health_dist	Numeric vector of health values across population groups.
pop_weights	Numeric vector of population weights (need not sum to 1; will be normalised internally).
epsilon	Inequality aversion parameter (default = 1). Must be non-negative. When $\varepsilon = 0$, returns 0 (no aversion).

Value

Atkinson index value in $[0, 1]$. A value of 0 indicates perfect equality; a value approaching 1 indicates maximum inequality.

References

Atkinson AB (1970). On the Measurement of Inequality. *Journal of Economic Theory* 2(3): 244-263. doi:[10.1016/00220531\(70\)900396](https://doi.org/10.1016/00220531(70)900396)

Examples

```
# Perfect equality
calc_atkinson_index(rep(70, 5), rep(0.2, 5), epsilon = 1)

# Gradient across groups
calc_atkinson_index(c(60, 63, 66, 69, 72), rep(0.2, 5), epsilon = 1)
```

`calc_concentration_index`*Calculate Concentration Index*

Description

Measures the degree to which a health variable is concentrated among socioeconomically advantaged or disadvantaged groups. A negative value indicates the health variable (e.g., illness) is concentrated among the deprived; a positive value indicates concentration among the advantaged.

Usage

```
calc_concentration_index(  
  data,  
  health_var,  
  group_var,  
  weight_var,  
  rank_var = NULL,  
  type = c("standard", "erreygers", "wagstaff")  
)
```

Arguments

<code>data</code>	A data frame with health and group columns.
<code>health_var</code>	Name of the health variable column (character).
<code>group_var</code>	Name of the socioeconomic group column (ordered integer, 1 = most deprived).
<code>weight_var</code>	Name of the population share column (sums to 1).
<code>rank_var</code>	Name of the socioeconomic rank variable (ridit scores, 0 = lowest, 1 = highest). If NULL, computed from <code>group_var</code> and <code>weight_var</code> using ridit scoring.
<code>type</code>	Concentration index variant: "standard" (Kakwani), "erreygers" (bounded), or "wagstaff" (normalised).

Value

A named list with `ci` (concentration index), `se`, and `type`.

References

Erreygers G (2009). Correcting the Concentration Index. *Journal of Health Economics* 28(2): 504-515. doi:10.1016/j.jhealeco.2008.02.003

Examples

```
df <- tibble::tibble(
  group      = 1:5,
  mean_hale  = c(60, 63, 66, 69, 72),
  pop_share  = rep(0.2, 5)
)
calc_concentration_index(df, "mean_hale", "group", "pop_share")
```

calc_edc

*Calculate Equally Distributed Equivalent Health (EDE)***Description**

Uses the Atkinson social welfare function to calculate EDE health — the level of health that, if equally distributed, would generate the same social welfare as the actual distribution given inequality aversion parameter η .

Usage

```
calc_edc(health_dist, pop_weights, eta = 1)
```

Arguments

health_dist Numeric vector of health values by group (must be strictly positive).

pop_weights Numeric vector of population weights (will be normalised to sum to 1).

eta Inequality aversion parameter (numeric scalar, default = 1).

- $\eta = 0$: returns arithmetic mean (no inequality aversion).
- $\eta = 1$: returns geometric mean (moderate aversion).
- $\eta > 1$: increasing inequality aversion.
- NICE relevant range: 0 to 10.

Details

$$\text{EDE}(\eta) = \left(\frac{\sum_i w_i h_i^{1-\eta}}{\sum_i w_i} \right)^{1/(1-\eta)} \quad \text{for } \eta \neq 1$$

$$\text{EDE}(1) = \exp \left(\sum_i w_i \ln(h_i) \right) \quad (\text{geometric mean, } \eta = 1)$$

Value

EDE health value (numeric scalar). Returns NA with a warning if any health values are non-positive.

References

Atkinson AB (1970). On the Measurement of Inequality. *Journal of Economic Theory* 2(3): 244-263. doi:10.1016/00220531(70)900396

Examples

```
health <- c(60, 63, 66, 69, 72)
weights <- rep(0.2, 5)

# eta = 0: arithmetic mean
calc_edc(health, weights, eta = 0)

# eta = 1: geometric mean
calc_edc(health, weights, eta = 1)

# eta = 5: high inequality aversion
calc_edc(health, weights, eta = 5)
```

calc_edc_profile	<i>Calculate EDE over a range of eta values</i>
------------------	---

Description

Evaluates `calc_edc` across a vector of η values and returns a tidy tibble. This is the basis for EDE profile plots as described in the York DCEA handbook.

Usage

```
calc_edc_profile(health_dist, pop_weights, eta_range = seq(0, 10, 0.1))
```

Arguments

health_dist	Numeric vector of health values by group.
pop_weights	Numeric vector of population weights.
eta_range	Numeric vector of η values to evaluate (default: <code>seq(0, 10, 0.1)</code>).

Value

A tibble with columns `eta` and `ede`.

Examples

```
health <- c(60, 63, 66, 69, 72)
weights <- rep(0.2, 5)
calc_edc_profile(health, weights)
```

`calc_equity_weight_profile`*Equity weights over a range of eta values*

Description

Convenience function that returns equity weights for each group across a range of η values. Useful for understanding how the choice of inequality aversion changes the implied weights.

Usage

```
calc_equity_weight_profile(  
  baseline_health,  
  pop_weights,  
  eta_range,  
  group_labels = NULL  
)
```

Arguments

<code>baseline_health</code>	Numeric vector of baseline health by group.
<code>pop_weights</code>	Numeric vector of population weights.
<code>eta_range</code>	Numeric vector of η values.
<code>group_labels</code>	Optional character vector of group labels.

Value

A tidy tibble with columns `eta`, `group`, `group_label` (if provided), and `equity_weight`.

Examples

```
baseline <- c(60, 63, 66, 69, 72)  
weights <- rep(0.2, 5)  
calc_equity_weight_profile(baseline, weights, eta_range = 0:5)
```

`calc_equity_weighted_nhb`*Calculate equity-weighted Net Health Benefit (NHB)*

Description

Applies equity weights to per-group NHB values to obtain the population-level equity-weighted NHB. This is the key summary statistic from the social welfare perspective.

Usage

```
calc_equity_weighted_nhb(nhb_by_group, equity_weights, pop_weights)
```

Arguments

`nhb_by_group` Numeric vector of net health benefit per group.
`equity_weights` Numeric vector of equity weights from `calc_equity_weights`.
`pop_weights` Numeric vector of population weights.

Value

Scalar equity-weighted NHB (numeric).

Examples

```
baseline <- c(60, 63, 66, 69, 72)
weights <- rep(0.2, 5)
ew <- calc_equity_weights(baseline, weights, eta = 1)
nhb <- c(100, 150, 200, 250, 300)
calc_equity_weighted_nhb(nhb, ew, weights)
```

`calc_equity_weights` *Calculate Equity Weights*

Description

Derives equity weights from the Atkinson social welfare function. Equity weights represent the relative social value of a one-unit health gain in each socioeconomic group given inequality aversion η .

Usage

```
calc_equity_weights(baseline_health, pop_weights, eta = 1, normalise = TRUE)
```

Arguments

`baseline_health` Numeric vector of baseline health (HALE) by group (ordered from most to least deprived).
`pop_weights` Numeric vector of population weights.
`eta` Inequality aversion parameter (default = 1).
`normalise` Logical. If TRUE (default), weights are normalised so their population-weighted mean equals 1. If FALSE, returns raw marginal welfare derivatives.

Details

For the Atkinson SWF, the equity weight for group i is proportional to $h_i^{-\eta}$: groups with lower baseline health receive higher weights when $\eta > 0$.

Value

Named numeric vector of equity weights, one per group.

References

Cookson R, Griffin S, Norheim OF, Culyer AJ (2020). Distributional Cost-Effectiveness Analysis. Oxford University Press. Oxford University Press (ISBN:9780198838197).

Robson M, Asaria M, Cookson R, Tsuchiya A, Ali S (2017). Eliciting the Level of Health Inequality Aversion in England. Health Economics 26(10): 1328-1334. doi:10.1002/hec.3386

Examples

```
baseline <- c(60, 63, 66, 69, 72)
weights <- rep(0.2, 5)
calc_equity_weights(baseline, weights, eta = 1)
```

calc_gini

Calculate Gini Coefficient for Health Distribution

Description

Computes the Gini coefficient as a measure of health inequality across socioeconomic groups. The Gini ranges from 0 (perfect equality) to 1 (maximum inequality).

Usage

```
calc_gini(health_dist, pop_weights = NULL)
```

Arguments

health_dist Numeric vector of health values (ordered from lowest to highest group).

pop_weights Optional numeric vector of population weights. If NULL, equal weights are assumed.

Value

Gini coefficient (numeric scalar in [0, 1]).

Examples

```
calc_gini(c(60, 63, 66, 69, 72), pop_weights = rep(0.2, 5))
```

calc_rii	<i>Calculate Relative Index of Inequality (RII)</i>
----------	---

Description

The RII is the SII expressed relative to the mean health level. An RII of 0.20 means the most deprived group has health 20 across the full socioeconomic range.

Usage

```
calc_rii(data, health_var, group_var, weight_var)
```

Arguments

data	A data frame with health and group columns.
health_var	Name of the health variable column (character).
group_var	Name of the socioeconomic group column (ordered integer, 1 = most deprived).
weight_var	Name of the population share column (sums to 1).

Value

A named list with elements rii, sii, se_rii, p_value, and model.

Examples

```
df <- tibble::tibble(
  group      = 1:5,
  mean_hale  = c(60, 63, 66, 69, 72),
  pop_share  = rep(0.2, 5)
)
calc_rii(df, "mean_hale", "group", "pop_share")
```

calc_sii	<i>Calculate Slope Index of Inequality (SII)</i>
----------	--

Description

Fits a weighted regression of health on ridit scores to estimate the absolute health difference between the most and least deprived groups. The SII is the regression coefficient on the ridit score, interpretable as the total health gap across the full socioeconomic range.

Usage

```
calc_sii(data, health_var, group_var, weight_var)
```

Arguments

<code>data</code>	A data frame with health and group columns.
<code>health_var</code>	Name of the health variable column (character).
<code>group_var</code>	Name of the socioeconomic group column (ordered integer, 1 = most deprived).
<code>weight_var</code>	Name of the population share column (sums to 1).

Value

A named list with elements:

- sii** Slope Index of Inequality (numeric)
- rii** Relative Index of Inequality (numeric)
- se_sii** Standard error of SII
- p_value** p-value for SII
- model** The underlying lm object

References

Mackenbach JP, Kunst AE (1997) Measuring the magnitude of socioeconomic inequalities in health: an overview of available measures illustrated with two examples from Europe. *Social Science and Medicine* 44(6): 757-771. doi:10.1016/S02779536(96)000731

Examples

```
df <- tibble::tibble(
  group      = 1:5,
  mean_hale  = c(60, 63, 66, 69, 72),
  pop_share  = rep(0.2, 5)
)
calc_sii(df, "mean_hale", "group", "pop_share")
```

calc_social_welfare *Calculate Social Welfare Function value*

Description

Wraps `calc_ede` to compute social welfare before and after an intervention and decomposes the welfare change into efficiency and equity components.

Usage

```
calc_social_welfare(baseline_health, post_health, pop_weights, eta = 1)
```

Arguments

baseline_health	Numeric vector of pre-intervention health by group.
post_health	Numeric vector of post-intervention health by group.
pop_weights	Numeric vector of population weights.
eta	Inequality aversion parameter (default = 1).

Value

A named list with elements:

ede_baseline	EDE health before intervention.
ede_post	EDE health after intervention.
delta_ede	Change in EDE (welfare gain).
efficiency_component	Change in mean health.
equity_component	Change in EDE minus change in mean.

Examples

```
pre <- c(60, 63, 66, 69, 72)
post <- c(61, 64, 66.5, 69.2, 72.1)
w <- rep(0.2, 5)
calc_social_welfare(pre, post, w, eta = 1)
```

canada_income_hale	<i>Canada income quintile HALE data</i>
--------------------	---

Description

Health-Adjusted Life Expectancy for Canada, stratified by household income quintile. For use in Canadian DCEA analyses (CADTH workflow).

Usage

```
canada_income_hale
```

Format

A tibble with 5 rows and 9 variables analogous to [england_imd_hale](#) but with income-based stratification.

Source

Statistics Canada, Health-Adjusted Life Expectancy by income quintile.

`compute_generalised_lorenz_data`*Compute Generalised Lorenz curve data*

Description

The Generalised Lorenz Curve (GLC) scales the Lorenz curve by mean health, making it sensitive to both inequality and average health level.

Usage

```
compute_generalised_lorenz_data(  
  health_dist,  
  pop_weights,  
  label = "Distribution"  
)
```

Arguments

<code>health_dist</code>	Numeric vector of health values by group.
<code>pop_weights</code>	Numeric vector of population weights.
<code>label</code>	Optional character label for this curve.

Value

A tibble with columns `cum_pop`, `cum_health_generalised`, and `label`.

Examples

```
compute_generalised_lorenz_data(c(60, 63, 66, 69, 72), rep(0.2, 5))
```

`compute_lorenz_data`*Compute Lorenz curve data*

Description

Returns the coordinates of the Lorenz curve for a health distribution. Groups are ordered from lowest to highest health (most to least deprived).

Usage

```
compute_lorenz_data(health_dist, pop_weights, label = "Distribution")
```

Arguments

health_dist	Numeric vector of health values by group.
pop_weights	Numeric vector of population weights.
label	Optional character label for this curve.

Value

A tibble with columns cum_pop (cumulative population share), cum_health (cumulative health share), and label.

Examples

```
compute_lorenz_data(c(60, 63, 66, 69, 72), rep(0.2, 5))
```

compute_ridit_scores *Compute ridit scores (cumulative midpoint ranks) for inequality measures*

Description

Ridit scores are used as the socioeconomic rank variable in concentration index calculations. For group i , the ridit is the cumulative population share up to the midpoint of group i .

Usage

```
compute_ridit_scores(pop_shares)
```

Arguments

pop_shares	Numeric vector of population proportions (ordered from most to least deprived, sums to 1).
------------	--

Value

Numeric vector of ridit scores in $[0, 1]$.

Examples

```
compute_ridit_scores(rep(0.2, 5))
```

england_imd_hale *England IMD quintile Health-Adjusted Life Expectancy data*

Description

Baseline HALE (Health-Adjusted Life Expectancy) at birth for England, stratified by Index of Multiple Deprivation (IMD) quintile. Quintile 1 is the most deprived; quintile 5 is the least deprived.

Usage

england_imd_hale

Format

A tibble with 5 rows and 14 variables:

imd_quintile Integer (1-5). 1 = most deprived.
group Integer (1-5). Standard group identifier (same as imd_quintile).
quintile_label Character. Human-readable quintile label.
group_label Character. Standard group label (same as quintile_label).
mean_hale Numeric. HALE at birth (years), both sexes (standard name).
mean_hale_all Numeric. HALE at birth (years), both sexes.
mean_hale_male Numeric. HALE at birth (years), males.
mean_hale_female Numeric. HALE at birth (years), females.
se_hale Numeric. Standard error of mean_hale (standard name).
se_hale_all Numeric. Standard error of mean_hale_all.
pop_share Numeric. Proportion of population in quintile (sums to 1).
cumulative_rank Numeric. Redit score for concentration index.
year Integer. Reference data year.
source Character. Data source.

Source

Office for Health Inequalities and Disparities (OHID) / Public Health England Health Profiles Plus. Proxy values based on published PHE data and interpolation from peer-reviewed literature.

References

Love-Koh J et al. (2019). Value in Health 22(5): 518-526. doi:10.1016/j.jval.2018.10.007

england_imd_qol	<i>England IMD quintile EQ-5D utility norms</i>
-----------------	---

Description

Age- and IMD-stratified EQ-5D-3L utility norms for England. Useful for assigning baseline quality of life weights in full-form DCEA.

Usage

england_imd_qol

Format

A tibble with 40 rows (5 IMD quintiles x 8 age bands) and 6 variables:

imd_quintile Integer (1-5).

age_band Character. Age band label.

mean_eq5d_utility Numeric. Mean EQ-5D-3L utility score.

se_eq5d Numeric. Standard error.

mean_qale_remaining Numeric. Quality-Adjusted Life Expectancy remaining (years).

source Character. Data source citation.

Source

Adapted from Ara R & Brazier JE (2010) with IMD gradient adjustments from Petrou et al. (Population Health Metrics).

example_cea_output	<i>Example CEA model output</i>
--------------------	---------------------------------

Description

A hypothetical cost-effectiveness analysis output for a lung cancer (NSCLC) treatment versus standard of care. Used in package examples and vignettes to demonstrate DCEA functions without requiring real data.

Usage

example_cea_output

Format

A list with two elements:

deterministic A tibble with columns: strategy, total_qaly, total_cost, inc_qaly, inc_cost, icer, nhb_at_20k, nhb_at_30k.

psa A data frame of 1000 PSA iterations with columns inc_qaly and inc_cost.

Source

Hypothetical data generated for illustration purposes only.

export_dcea_excel	<i>Export DCEA results to Excel (NICE submission format)</i>
-------------------	--

Description

Writes a multi-sheet Excel workbook with NICE-formatted DCEA output, including the summary table, per-group results, inequality indices, and social welfare profile.

Usage

```
export_dcea_excel(dcea_result, filepath, include_plots = FALSE)
```

Arguments

dcea_result Object of class "aggregate_dcea" or "full_dcea".

filepath Output .xlsx file path (character).

include_plots Logical. Embed plots as images in the Excel workbook (default: FALSE).

Value

Invisibly returns filepath.

Examples

```
result <- run_aggregate_dcea(  
  icer = 25000, inc_qaly = 0.5, inc_cost = 12500,  
  population_size = 10000, wtp = 20000  
)  
export_dcea_excel(result, file.path(tempdir()), "my_dcea_results.xlsx"))
```

generate_dcea_report *Generate a full DCEA report*

Description

Renders a complete DCEA report as an HTML, Word, or PDF document using R Markdown.

Usage

```
generate_dcea_report(  
  dcea_result,  
  format = "html",  
  filepath = NULL,  
  template = "nice_submission"  
)
```

Arguments

dcea_result	Object of class "aggregate_dcea" or "full_dcea".
format	Output format: "html" (default), "word", "pdf".
filepath	Output file path. If NULL, a temporary file is used and the report is opened in a browser.
template	Report template: "nice_submission" (default), "academic", "cadth".

Value

Invisibly returns the output file path.

Examples

```
result <- run_aggregate_dcea(  
  icer = 25000, inc_qaly = 0.5, inc_cost = 12500,  
  population_size = 10000, wtp = 20000  
)  
generate_dcea_report(result, format = "html")
```

generate_nice_table *Generate NICE-formatted DCEA submission table*

Description

Creates a summary table formatted according to NICE (2025) Methods Support Document guidance for DCEA as supplementary evidence in technology appraisals.

Usage

```
generate_nice_table(dcea_result, format = "tibble", include_psa = FALSE)
```

Arguments

dcea_result Object of class "aggregate_dcea" or "full_dcea".

format Output format: "tibble" (default), "flextable", "html", or "xlsx".

include_psa Logical. Include probabilistic uncertainty columns if PSA data are available (default: FALSE).

Value

A formatted table object of the requested type.

Examples

```
result <- run_aggregate_dcea(  
  icer = 25000, inc_qaly = 0.5, inc_cost = 12500,  
  population_size = 10000, wtp = 20000  
)  
generate_nice_table(result, format = "tibble")
```

get_baseline_health *Get baseline health distribution for a country*

Description

Returns pre-loaded HALE (Health-Adjusted Life Expectancy) data stratified by equity subgroup for use in DCEA. Data are sourced from ONS/OHID (England), Statistics Canada, and the WHO Global Health Observatory.

Usage

```
get_baseline_health(
  country = "england",
  equity_var = "imd_quintile",
  age_group = "all",
  sex = "all",
  year = NULL
)
```

Arguments

country	Character. One of "england", "canada", "who_regions", "scotland", "wales".
equity_var	Character. Stratification variable. Options depend on country: <ul style="list-style-type: none"> • England: "imd_quintile" (default), "imd_decile" • Canada: "income_quintile" • WHO: "who_region"
age_group	Character. Age filter (default "all").
sex	Character. Sex filter: "all" (default), "male", "female".
year	Integer. Data year. Uses most recent available if NULL.

Value

A tibble with columns: group, group_label, mean_hale, se_hale, pop_share, cumulative_rank, source, year.

Examples

```
england_baseline <- get_baseline_health("england", "imd_quintile")
england_baseline
```

```
merge_cea_with_baseline
```

Merge CEA model output with baseline health distribution

Description

Joins a CEA result data frame (one row per equity subgroup) with a baseline health distribution returned by [get_baseline_health](#). This is the key data-preparation step before running DCEA.

Usage

```
merge_cea_with_baseline(cea_output, baseline, by = "group")
```

Arguments

cea_output	Data frame of CEA results with at least one column matching the baseline by variable.
baseline	Tibble returned by <code>get_baseline_health</code> .
by	Column name to join on (default: "group").

Value

A merged tibble suitable for `run_full_dcea`.

Examples

```
baseline <- get_baseline_health("england", "imd_quintile")
cea_out <- tibble::tibble(
  group = 1:5,
  inc_qaly = c(0.3, 0.4, 0.5, 0.55, 0.6),
  inc_cost = rep(10000, 5)
)
merge_cea_with_baseline(cea_out, baseline, by = "group")
```

normalise_weights *Normalise population weights to sum to 1*

Description

Normalise population weights to sum to 1

Usage

```
normalise_weights(weights)
```

Arguments

weights	Numeric vector of weights.
---------	----------------------------

Value

Normalised numeric vector.

Examples

```
normalise_weights(c(1, 2, 3, 4, 5))
```

nsclc_dcea_example	<i>NSCLC DCEA worked example</i>
--------------------	----------------------------------

Description

A full DCEA worked example based on published literature for a non-small-cell lung cancer treatment. Includes subgroup-level CEA results by IMD quintile for use in full-form DCEA demonstrations.

Usage

```
nsclc_dcea_example
```

Format

A list with elements:

subgroup_cea Tibble of per-IMD-quintile CEA results.

baseline Baseline health distribution for NSCLC population.

staircase Staircase data for the inequality staircase plot.

Source

Adapted from published NSCLC DCEA literature for illustration.

plot.aggregate_dcea	<i>Plot method for aggregate_dcea</i>
---------------------	---------------------------------------

Description

Dispatches to [plot_equity_impact_plane](#) by default.

Usage

```
## S3 method for class 'aggregate_dcea'
plot(x, type = "impact_plane", ...)
```

Arguments

x	An object of class "aggregate_dcea".
type	Plot type: "impact_plane" (default), "lorenz", or "ede_profile".
...	Additional arguments passed to the underlying plot function.

Value

A **ggplot2** object.

plot.full_dcea	<i>Plot method for full_dcea</i>
----------------	----------------------------------

Description

Plot method for full_dcea

Usage

```
## S3 method for class 'full_dcea'
plot(x, type = "impact_plane", ...)
```

Arguments

x	An object of class "full_dcea".
type	Plot type (default "impact_plane").
...	Additional arguments passed to the underlying plot function.

Value

A **ggplot2** object.

plot_dcea_tornado	<i>Plot Tornado Diagram for DCEA Sensitivity Analysis</i>
-------------------	---

Description

Creates a tornado diagram showing the influence of each parameter on the net health benefit. Parameters are sorted by range (most influential at the top).

Usage

```
plot_dcea_tornado(sensitivity_result)
```

Arguments

sensitivity_result	Output from run_dcea_sensitivity .
--------------------	--

Value

A **ggplot2** tornado diagram.

Examples

```
result <- run_aggregate_dcea(  
  icer = 25000, inc_qaly = 0.5, inc_cost = 12500,  
  population_size = 10000, wtp = 20000  
)  
sa <- run_dcea_sensitivity(result)  
plot_dcea_tornado(sa)
```

plot_edc_profile *Plot EDE Profile*

Description

Shows how equity-weighted NHB (or EDE health) changes as inequality aversion (η) increases. The profile reveals the critical η at which an intervention becomes welfare-improving after accounting for equity concerns.

Usage

```
plot_edc_profile(  
  dcea_result,  
  eta_range = seq(0, 15, 0.1),  
  comparators = NULL,  
  show_benchmark_eta = TRUE  
)
```

Arguments

dcea_result DCEA result object.

eta_range Numeric vector of η values (default: seq(0, 15, 0.1)).

comparators Optional list of additional DCEA result objects to overlay on the same plot.

show_benchmark_eta
 Logical. Mark commonly used η values (0 = efficiency only; 1 = Robson et al. UK estimate; 10 = strong aversion; default: TRUE).

Value

A **ggplot2** object.

Examples

```
result <- run_aggregate_dcea(  
  icer = 25000, inc_qaly = 0.5, inc_cost = 12500,  
  population_size = 10000, wtp = 20000  
)  
plot_edc_profile(result)
```

 plot_equity_impact_plane

Plot Equity-Efficiency Impact Plane

Description

Creates the equity-efficiency impact plane as described in Cookson et al. (2017) Value in Health. The x-axis shows health inequality impact (change in a chosen inequality index), the y-axis shows net health benefit (efficiency). Four quadrants represent: Win-Win (NE), equity gain and efficiency loss (NW), equity loss and efficiency gain (SE), Lose-Lose (SW).

Usage

```
plot_equity_impact_plane(
  dcea_result,
  comparators = NULL,
  x_axis = "sii_change",
  y_axis = "nhb",
  show_psa_cloud = TRUE,
  show_quadrant_labels = TRUE,
  show_threshold_lines = TRUE,
  point_labels = NULL,
  colour_palette = NULL,
  theme_style = "publication"
)
```

Arguments

dcea_result	Object of class "aggregate_dcea" or "full_dcea".
comparators	Optional list of additional DCEA result objects to overlay on the same plane (for multi-comparator plots).
x_axis	Inequality metric for x-axis. One of "sii_change" (default), "atkinson_change", "gini_change".
y_axis	Health outcome for y-axis. One of "nhb" (default), "net_monetary_benefit".
show_psa_cloud	Logical. Show probabilistic scatter cloud if PSA data are available (default: TRUE).
show_quadrant_labels	Logical (default: TRUE).
show_threshold_lines	Logical. Show NHB = 0 and inequality = 0 reference lines (default: TRUE).
point_labels	Optional character vector of labels for points.
colour_palette	Optional named character vector of hex colours.
theme_style	Visual theme: "publication" (default) or "ggplot_default".

Value

A **ggplot2** object.

References

Cookson R, Asaria M, Ali S, Shaw R, Doran T, Goldblatt P (2017). Health equity monitoring for healthcare quality assurance. *Social Science & Medicine* 198: 148-156.

Examples

```
result <- run_aggregate_dcea(
  icer = 25000, inc_qaly = 0.5, inc_cost = 12500,
  population_size = 10000, wtp = 20000
)
plot_equity_impact_plane(result)
```

plot_inequality_staircase

Plot Inequality Staircase

Description

Visualises the causal pathway from intervention access to health inequality impact across the five staircase steps: (1) disease prevalence by group, (2) eligibility, (3) uptake/access, (4) clinical effect, (5) opportunity cost distribution.

Usage

```
plot_inequality_staircase(staircase_data, equity_var = "imd_quintile")
```

Arguments

`staircase_data` Data frame with columns: `step` (integer 1-5), `step_label` (character), `group` (integer), `group_label` (character), `value` (numeric).

`equity_var` Equity stratification variable name (for axis label).

Value

A **ggplot2** faceted plot.

Examples

```
staircase_df <- data.frame(
  step      = rep(1:5, each = 5),
  step_label = rep(c("Prevalence", "Eligibility", "Uptake",
                    "Clinical effect", "Opportunity cost"), each = 5),
  group     = rep(1:5, times = 5),
  group_label = rep(paste("Q", 1:5), times = 5),
```

```

value      = c(0.30, 0.28, 0.25, 0.22, 0.18,
               0.90, 0.88, 0.85, 0.82, 0.80,
               0.70, 0.65, 0.60, 0.55, 0.50,
               0.45, 0.44, 0.43, 0.42, 0.40,
               0.20, 0.18, 0.17, 0.15, 0.12)
)
plot_inequality_staircase(staircase_df)

```

plot_lorenz_curve *Plot Lorenz Curve for Health Distribution*

Description

Plots the Lorenz curve showing health concentration across the socioeconomic distribution. The 45-degree line represents perfect equality.

Usage

```
plot_lorenz_curve(dcea_result, show_pre_post = TRUE, show_generalised = FALSE)
```

Arguments

dcea_result DCEA result object, or a data frame with health and weight columns.

show_pre_post Logical. Overlay pre- and post-intervention curves (default: TRUE).

show_generalised Logical. Overlay the Generalised Lorenz Curve (default: FALSE).

Value

A **ggplot2** object.

Examples

```

result <- run_aggregate_dcea(
  icer = 25000, inc_qaly = 0.5, inc_cost = 12500,
  population_size = 10000, wtp = 20000
)
plot_lorenz_curve(result)

```

prepare_subgroup_cea *Prepare subgroup CEA data for DCEA*

Description

Validates and standardises a data frame of subgroup-specific CEA results for use in [run_full_dcea](#).

Usage

```
prepare_subgroup_cea(  
  data,  
  group_var,  
  inc_qaly_var,  
  inc_cost_var,  
  pop_share_var  
)
```

Arguments

data	Data frame with per-subgroup CEA results.
group_var	Name of the group identifier column.
inc_qaly_var	Name of the incremental QALY column.
inc_cost_var	Name of the incremental cost column.
pop_share_var	Name of the population share column.

Value

A validated and normalised tibble.

Examples

```
df <- tibble::tibble(  
  group = 1:5,  
  inc_qaly = c(0.3, 0.4, 0.5, 0.55, 0.6),  
  inc_cost = rep(10000, 5),  
  pop_share = rep(0.2, 5)  
)  
prepare_subgroup_cea(df, "group", "inc_qaly", "inc_cost", "pop_share")
```

print.aggregate_dcea *Print method for aggregate_dcea*

Description

Print method for aggregate_dcea

Usage

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

Arguments

x An object of class "aggregate_dcea".
... Further arguments (ignored).

Value

Invisibly returns x.

print.full_dcea *Print method for full_dcea*

Description

Print method for full_dcea

Usage

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

Arguments

x An object of class "full_dcea".
... Further arguments (ignored).

Value

Invisibly returns x.

run_aggregate_dcea *Run Aggregate DCEA*

Description

Implements the aggregate DCEA approach of Love-Koh et al. (2019) Value in Health. Uses disease-level healthcare utilisation patterns to distribute average health benefits from a standard CEA across socioeconomic groups. This is the method supported by NICE (2025) as a supplementary analysis for technology appraisals.

Usage

```
run_aggregate_dcea(
  icer,
  inc_qaly,
  inc_cost,
  population_size,
  wtp = 20000,
  disease_icd = NULL,
  subgroup_distribution = NULL,
  baseline_health = NULL,
  equity_var = "imd_quintile",
  wtp_for_equity = NULL,
  opportunity_cost_threshold = 13000,
  psa_results = NULL
)
```

Arguments

icer	Incremental cost-effectiveness ratio (GBP per QALY).
inc_qaly	Incremental QALYs per patient (from base-case CEA).
inc_cost	Incremental cost per patient (from base-case CEA).
population_size	Total eligible population size (integer).
wtp	Willingness-to-pay threshold in GBP/QALY (default: 20000).
disease_icd	ICD-10 code or description for HES utilisation lookup. Used to distribute benefits across IMD groups if subgroup_distribution is NULL. Example: "C34" for lung cancer.
subgroup_distribution	Optional named numeric vector (length = number of equity subgroups) giving the proportion of patients in each group. Names should match group labels in the baseline dataset. Must sum to 1. If NULL, derived from disease_icd via internal lookup.
baseline_health	Optional tibble from get_baseline_health . If NULL, uses England IMD quintile data.

`equity_var` Equity stratification variable (default: "imd_quintile").

`wtp_for_equity` Optional second WTP threshold for equity-weighted analysis.

`opportunity_cost_threshold`
Cost per QALY of care displaced by the intervention's budget impact (default: 13000, i.e., NICE's k threshold).

`psa_results` Optional data frame of PSA iteration results (one row per iteration, columns `inc_qaly` and `inc_cost`).

Value

An object of class "aggregate_dcea", a named list with:

`summary` Key scalar DCEA outputs.

`by_group` Per-group tibble: health gain, opportunity cost, NHB.

`inequality_impact` Pre/post inequality indices.

`social_welfare` Social welfare results over eta.

`equity_plane_data` Data frame for `plot_equity_impact_plane`.

`metadata` Inputs and assumptions.

References

Love-Koh J, Asaria M, Cookson R, Griffin S (2019). The Social Distribution of Health: Estimating Quality-Adjusted Life Expectancy in England. *Value in Health* 22(5): 518-526. doi:10.1016/j.jval.2018.10.007

See Also

[plot_equity_impact_plane](#), [run_full_dcea](#)

Examples

```
result <- run_aggregate_dcea(
  icer           = 25000,
  inc_qaly       = 0.5,
  inc_cost       = 12500,
  population_size = 10000,
  wtp            = 20000
)
summary(result)
```

 run_dcea_sensitivity *Run DCEA Sensitivity Analysis*

Description

Performs systematic one-way and multi-way sensitivity analysis across key DCEA parameters: inequality aversion (η), WTP threshold, opportunity cost threshold, subgroup distribution assumptions, and equity measure choice.

Usage

```
run_dcea_sensitivity(
  dcea_result,
  params_to_vary = "all",
  eta_range = 0:10,
  wtp_range = NULL,
  occ_range = NULL
)
```

Arguments

dcea_result	Object of class "aggregate_dcea" or "full_dcea".
params_to_vary	Character vector of parameter names to vary. Options: "eta", "wtp", "occ_threshold", "subgroup_distribution", "equity_measure", "all" (default: "all").
eta_range	Numeric vector of η values for eta sensitivity (default: 0:10).
wtp_range	Numeric vector of WTP values to test (default: varies $\pm 50\%$ around base case).
occ_range	Numeric vector of opportunity cost threshold values (default: c(8000, 10000, 13000, 15000, 20000)).

Value

An object of class "dcea_sensitivity" with elements:

eta_profile Tibble: NHB and SII change across eta range.
 one_way Tibble: results of one-way sensitivity for all parameters.
 tornado_data Data frame ready for tornado plot.
 parameters List of parameter ranges used.

Examples

```
result <- run_aggregate_dcea(
  icer = 25000, inc_qaly = 0.5, inc_cost = 12500,
  population_size = 10000, wtp = 20000
)
sa <- run_dcea_sensitivity(result, params_to_vary = "eta")
plot_dcea_tornado(sa)
```

run_full_dcea	<i>Run Full-Form DCEA</i>
---------------	---------------------------

Description

Implements full-form DCEA where subgroup-specific model parameters are available (e.g., differential uptake, differential quality-of-life gains, differential survival by socioeconomic group). Full-form DCEA is appropriate when:

- The disease has well-documented SES gradients in clinical outcomes.
- Subgroup trial data or real-world evidence is available.
- NICE requires more granular equity evidence (HST or exceptional cases).

Usage

```
run_full_dcea(
  subgroup_cea_results,
  baseline_health,
  wtp = 20000,
  opportunity_cost_threshold = 13000,
  uptake_by_group = NULL,
  adherence_by_group = NULL,
  comorbidity_adjustment = FALSE,
  psa_iterations = NULL
)
```

Arguments

subgroup_cea_results	Data frame with one row per equity subgroup. Required columns: group (integer), group_label (character), inc_qaly (numeric), inc_cost (numeric), pop_share (numeric, sums to 1).
baseline_health	Tibble from get_baseline_health .
wtp	Willingness-to-pay threshold in GBP/QALY (default: 20000).
opportunity_cost_threshold	Cost per QALY of displaced care (default: 13000).
uptake_by_group	Optional named numeric vector of uptake rates (0-1) per group. If NULL, assumes equal uptake across all groups.
adherence_by_group	Optional named numeric vector of adherence rates (0-1) per group. Applied as a multiplier on inc_qaly.
comorbidity_adjustment	Logical. If TRUE, applies an SES-based comorbidity adjustment to QoL gains (experimental; default: FALSE).

`psa_iterations` Optional integer. Number of PSA iterations. If provided, returns probabilistic NHB distribution by group.

Value

An object of class "full_dcea" with elements analogous to [run_aggregate_dcea](#) plus subgroup-level detail.

Examples

```
baseline <- get_baseline_health("england", "imd_quintile")
subgroup_data <- tibble::tibble(
  group      = 1:5,
  group_label = paste("IMD Q", 1:5),
  inc_qaly    = c(0.30, 0.38, 0.45, 0.52, 0.58),
  inc_cost    = c(12000, 11500, 11000, 10500, 10000),
  pop_share   = rep(0.2, 5)
)
result <- run_full_dcea(subgroup_data, baseline)
summary(result)
```

summary.aggregate_dcea

Summary method for aggregate_dcea

Description

Summary method for `aggregate_dcea`

Usage

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

Arguments

`object` An object of class "aggregate_dcea".
`...` Further arguments (ignored).

Value

A tibble of key DCEA outputs.

summary.full_dcea	<i>Summary method for full_dcea</i>
-------------------	-------------------------------------

Description

Summary method for full_dcea

Usage

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

Arguments

object	An object of class "full_dcea".
...	Further arguments (ignored).

Value

Invisibly returns object.

validate_dcea_data	<i>Validate and format DCEA input data</i>
--------------------	--

Description

Checks that a data frame has required columns, correct types, and that population weights sum to 1. Returns the data frame invisibly if valid, or throws an informative error.

Usage

```
validate_dcea_data(data, required_cols, weight_var, tol = 1e-06)
```

Arguments

data	A data frame to validate.
required_cols	Character vector of required column names.
weight_var	Name of the population weight column.
tol	Tolerance for checking that weights sum to 1 (default: 1e-6).

Value

The input data frame, invisibly.

Examples

```
df <- tibble::tibble(  
  group      = 1:5,  
  mean_hale  = c(60, 63, 66, 69, 72),  
  pop_share  = rep(0.2, 5)  
)  
validate_dcea_data(df, c("group", "mean_hale", "pop_share"), "pop_share")
```

who_regions_hale	<i>WHO regional HALE data</i>
------------------	-------------------------------

Description

Health-Adjusted Life Expectancy at birth for the six WHO regions. Useful for international equity analyses and global burden of disease perspectives.

Usage

```
who_regions_hale
```

Format

A tibble with 6 rows and 8 variables:

who_region Character. WHO region code.

region_label Character. Full region name.

mean_hale Numeric. HALE at birth (years).

se_hale Numeric. Standard error.

pop_share Numeric. Proportion of world population.

cumulative_rank Numeric. Redit score.

year Integer. Reference year.

source Character. WHO GHO data citation.

Source

WHO Global Health Observatory. <https://www.who.int/data/gho>

Index

* datasets

- canada_income_hale, 15
 - england_imd_hale, 18
 - england_imd_qol, 19
 - example_cea_output, 19
 - nsclc_dcea_example, 25
 - who_regions_hale, 39
- build_staircase_data, 4
- calc_all_inequality_indices, 5
- calc_atkinson_index, 6
- calc_concentration_index, 7
- calc_edc, 3, 8, 9, 14
- calc_edc_profile, 9
- calc_equity_weight_profile, 10
- calc_equity_weighted_nhb, 10
- calc_equity_weights, 11, 11
- calc_gini, 12
- calc_rii, 3, 13
- calc_sii, 3, 13
- calc_social_welfare, 14
- canada_income_hale, 15
- compute_generalised_lorenz_data, 16
- compute_lorenz_data, 16
- compute_ridit_scores, 17
- dceasimR (dceasimR-package), 3
- dceasimR-package, 3
- england_imd_hale, 15, 18
- england_imd_qol, 19
- example_cea_output, 19
- export_dcea_excel, 20
- generate_dcea_report, 21
- generate_nice_table, 22
- get_baseline_health, 3, 22, 23, 24, 33, 36
- merge_cea_with_baseline, 23
- normalise_weights, 24
- nsclc_dcea_example, 25
- plot.aggregate_dcea, 25
- plot.full_dcea, 26
- plot_dcea_tornado, 26
- plot_edc_profile, 27
- plot_equity_impact_plane, 3, 25, 28, 34
- plot_inequality_staircase, 4, 29
- plot_lorenz_curve, 30
- prepare_subgroup_cea, 31
- print.aggregate_dcea, 32
- print.full_dcea, 32
- run_aggregate_dcea, 3, 33, 37
- run_dcea_sensitivity, 26, 35
- run_full_dcea, 3, 24, 31, 34, 36
- summary.aggregate_dcea, 37
- summary.full_dcea, 38
- validate_dcea_data, 38
- who_regions_hale, 39