Package: DUToolkit (via r-universe)

June 3, 2026

Title Visualizing and Quantifying Decision Uncertainty

Version 1.0.2

Description A suite of tools to help modelers and decision-makers effectively interpret and communicate decision risk when evaluating multiple policy options. It uses model outputs from uncertainty analysis for baseline scenarios and policy alternatives to generate visual representations of uncertainty and quantitative measures for assessing associated risks. For more details see Wiggins and colleagues (2025) <doi:10.1371/journal.pone.0332522> and <https://dut.ihe.ca/>.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.1

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

VignetteBuilder knitr

Depends R (>= 3.5.0)

LazyData true

LazyDataCompression xz

Imports ggdist, ggplot2, ggtext, rlang, scales, stats

Config/testthat/edition 3

NeedsCompilation no

Author Megan Wiggins [aut, cre] (ORCID: <https://orcid.org/0009-0007-0290-4950>), Marie Betsy Varughese [aut] (ORCID: <https://orcid.org/0000-0002-8432-6539>), Ellen Rafferty [aut] (ORCID: <https://orcid.org/0000-0001-9974-8016>), Sasha van Katwyk [aut] (ORCID: <https://orcid.org/0000-0003-3026-2063>), Christopher McCabe [aut] (ORCID: <https://orcid.org/0000-0001-5728-4129>), Jeff Round [aut] (ORCID: <https://orcid.org/0000-0002-3103-6984>), Erin Kirwin [aut] (ORCID: <https://orcid.org/0000-0001-6957-7236>), Institute of Health Economics [cph, aut], Canadian Network for Modelling Infectious Diseases [fnd]

Maintainer Megan Wiggins <mwiggins@ihe.ca>

Config/pak/sysreqs libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev

Repository https://mwiggins-ihe.r-universe.dev

Date/Publication 2025-10-06 16:40:02 UTC

RemoteUrl https://github.com/cran/DUToolkit

RemoteRef HEAD

RemoteSha 55ef2dbd97043fbccb4881e70017d56199389225

Contents

`calculate_max_min_risk`

Calculate risk measures at peak (or minimum)

Description

For each policy alternative, this function calculates the risk measure at the peak values (or lowest values if the threshold is a minimum).

Usage

```
calculate_max_min_risk(max_min_values_list, D, Dt_max = TRUE)
```

Arguments

`max_min_values_list`	A list generated by `get_max_min_values()`
`D`	A single threshold value
`Dt_max`	A logical value indicating whether the decision threshold is a maximum (`TRUE`) or a minimum (`FALSE`). The default is `TRUE`.

Value

A list of risk measure values.

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
D <- 750

peak_values_list <- get_max_min_values(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt_max = TRUE
)

calculate_max_min_risk(
  peak_values_list,
  D = D,
  Dt_max = TRUE
)
```

---

calculate_risk *Calculate risk measures*

Description

This function calculates the expected risk measure for each policy alternative using the outputs from multiple model runs with different input parameter sets (e.g., probabilistic sensitivity, uncertainty, or Bayesian inference analysis).

Usage

```
calculate_risk(data, tmin, tmax, Dt, Dt_max = TRUE, W = FALSE, weight = NULL)
```

Arguments

`data`	A list of data.frames (one data.frame for each policy alternative).
`tmin`	A numeric value or a date specifying the minimum simulation time to include in the analysis (ex. 1 or "2021-01-01"). This should correspond to a value in the first column of each data.frame in `data`.
`tmax`	A numeric value or a date specifying the maximum simulation time to include in the analysis (ex. 100 or "2021-04-10"). This should correspond to a value in the first column of each data.frame in `data`.
`Dt`	A numeric vector of decision thresholds, one for each model time step between `tmin` and `tmax`.

Dt_max	A logical value indicating whether the decision threshold is a maximum (TRUE) or a minimum (FALSE). The default is TRUE.
W	A logical value indicating whether the risk calculation should be weighted (TRUE) or not (FALSE). The default is FALSE (an equal weight of 1 is assumed for all simulation runs).
weight	A list of numeric vectors of weights for the risk calculation, with one weight corresponding to each time step. Required if W is TRUE.

Value

A list of risk scores, one for each policy alternative.

data format

Each data.frame in data contains the results from multiple model runs using different parameter sets (e.g., from probabilistic sensitivity, uncertainty, or Bayesian inference analysis). The first column contains the model time and subsequent columns contain the predicted output for each simulation run at the respective time. The model time in the first column must contain numeric values indicating a simulation time (ex. 1, 2, 3,...) or dates (ex. "2021-01-01", "2021-01-02") which must be in R Date format (i.e., class="Date"). To ensure a consistent basis for comparison, the model time in the first column should be the same for each policy alternative (data.frame).

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
Dt <- rep(750, nrow(psa_data$Baseline))

calculate_risk(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt = Dt,
  Dt_max = TRUE
)
```

---

calculate_threshold_probs

Calculate threshold exceedance probabilities

Description

For each policy alternative, this function calculates the probability that the peak (or minimum) value exceeds (or is below) a specified threshold(s) using a Riemann sum approach.

Usage

```
calculate_threshold_probs(max_min_values_list, Dp, Dt_max = TRUE)
```

Arguments

max_min_values_list	A list generated by get_max_min_values()
Dp	A vector of threshold values to calculate the probabilities for.
Dt_max	A logical value indicating whether the decision threshold is a maximum (TRUE) or a minimum (FALSE). The default is TRUE.

Value

A list of vectors of threshold values and corresponding probabilities.

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
D <- 750
Dp <- c(750, 1000, 2000)

peak_values_list <- get_max_min_values(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt_max = TRUE
)

calculate_threshold_probs(
  peak_values_list,
  Dp = Dp,
  Dt_max = TRUE
)
```

---

calculate_time *Calculate time of threshold exceedance*

Description

For each policy alternative, this function calculates: (i) the percent of simulations in which the threshold is exceeded (or not met if the threshold is a minimum), (ii) the mean simulation time of the first exceedance and 95th percentile range, (iii) the mean duration of the first exceedance and 95th percentile range, and (iv) if the first column of model output data.frame passed to the function is a Date, the mean date of the first and last exceedance.

Usage

```
calculate_time(data, tmin, tmax, Dt, Dt_max = TRUE)
```

Arguments

data	A list of data.frames (one data.frame for each policy alternative).
tmin	A numeric value or a date specifying the minimum simulation time to include in the analysis (ex. 1 or "2021-01-01"). This should correspond to a value in the first column of each data.frame in data.
tmax	A numeric value or a date specifying the maximum simulation time to include in the analysis (ex. 100 or "2021-04-10"). This should correspond to a value in the first column of each data.frame in data.
Dt	A numeric vector of decision thresholds, one for each model time step between tmin and tmax.
Dt_max	A logical value indicating whether the decision threshold is a maximum (TRUE) or a minimum (FALSE). The default is TRUE.

Value

A list of data.frames, one for each policy alternative.

data format

Each data.frame in data contains the results from multiple model runs using different parameter sets (e.g., from probabilistic sensitivity, uncertainty, or Bayesian inference analysis). The first column contains the model time and subsequent columns contain the predicted output for each simulation run at the respective time. The model time in the first column must contain numeric values indicating a simulation time (ex. 1, 2, 3,...) or dates (ex. "2021-01-01", "2021-01-02") which must be in R Date format (i.e., class="Date"). To ensure a consistent basis for comparison, the model time in the first column should be the same for each policy alternative (data.frame).

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
Dt <- rep(750, nrow(psa_data$Baseline))

calculate_time(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt = Dt,
  Dt_max = TRUE
)
```

`get_max_min_values` *Finds peak (or lowest) model output values*

Description

For each policy alternative, this function finds the peak (or lowest if the threshold is a minimum) model output values for each simulation run.

Usage

```
get_max_min_values(data, tmin, tmax, Dt_max = TRUE)
```

Arguments

`data`	A list of data.frames (one data.frame for each policy alternative).
`tmin`	A numeric value or a date specifying the minimum simulation time to include in the analysis (ex. 1 or "2021-01-01"). This should correspond to a value in the first column of each data.frame in `data`.
`tmax`	A numeric value or a date specifying the maximum simulation time to include in the analysis (ex. 100 or "2021-04-10"). This should correspond to a value in the first column of each data.frame in `data`.
`Dt_max`	A logical value indicating whether the decision threshold is a maximum (`TRUE`) or a minimum (`FALSE`). The default is `TRUE`.

Value

A list of data.frame(s) containing the peak (or lowest) value and corresponding simulation time for each policy alternative.

data format

Each data.frame in `data` contains the results from multiple model runs using different parameter sets (e.g., from probabilistic sensitivity, uncertainty, or Bayesian inference analysis). The first column contains the model time and subsequent columns contain the predicted output for each simulation run at the respective time. The model time in the first column must contain numeric values indicating a simulation time (ex. 1, 2, 3,...) or dates (ex. "2021-01-01", "2021-01-02") which must be in `R` Date format (i.e., class="Date"). To ensure a consistent basis for comparison, the model time in the first column should be the same for each policy alternative (data.frame).

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
Dt <- rep(750, nrow(psa_data$Baseline))

peak_values_list <- get_max_min_values(
  psa_data,
```

```
  tmin = tmin,
  tmax = tmax,
  Dt_max = TRUE
)
```

`get_relative_values` *Find output values at time points relative to the peak (or minimum)*

Description

For each policy alternative, this function finds the the model output values for each simulation run at the specified time points relative to the peak (or minimum) value.

Usage

```
get_relative_values(data, max_min_values_list, t_s, t_ss, Dt_max = TRUE)
```

Arguments

`data`	A list of data.frames (one data.frame for each policy alternative).
`max_min_values_list`	A list generated by `get_max_min_values()`.
`t_s`	The total time window to examine before and after the peak (or minimum) value.
`t_ss`	The interval size for sampling time points within the specified time window `t_s`. For example, if `t_s = 20` and `t_ss = 10` the function will extract values at peak_time-20, peak_time-10, peak_time, peak_time+10, peak_time+20.
`Dt_max`	A logical value indicating whether the decision threshold is a maximum (`TRUE`) or a minimum (`FALSE`). The default is `TRUE`.

Value

A list of nested lists with the following elements: (i) a data.frame of values recorded at specific time steps relative to the peak time (e.g., peak-10, peak, peak+10) for each model run, and (ii) a vector of the names of each time step ex. minus_30,minus_20,...,plus_20, plus_30.

data format

Each data.frame in `data` contains the results from multiple model runs using different parameter sets (e.g., from probabilistic sensitivity, uncertainty, or Bayesian inference analysis). The first column contains the model time and subsequent columns contain the predicted output for each simulation run at the respective time. The model time in the first column must contain numeric values indicating a simulation time (ex. 1, 2, 3,...) or dates (ex. "2021-01-01", "2021-01-02") which must be in `R` Date format (i.e., class="Date"). To ensure a consistent basis for comparison, the model time in the first column should be the same for each policy alternative (data.frame).

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
D <- 750
t_s <- 20
t_ss <- 10

peak_values_list <- get_max_min_values(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt_max = TRUE
)

peak_temporal_list <- get_relative_values(
  psa_data,
  peak_values_list,
  t_s = t_s,
  t_ss = t_ss
)
```

`plot_density` *Generate density plots*

Description

For each policy alternative, this function generates probability density plots of the highest (or lowest if the threshold is a minimum) projected outcome across simulation runs. The decision threshold is shown directly on the plot as a vertical line. The area under the probability density curve where the threshold value is exceeded is shaded to visually display the downside risk of the policy alternative.

Usage

```
plot_density(max_min_values_list, D, Dt_max = TRUE, risk_measures)
```

Arguments

`max_min_values_list`	A list generated by `get_max_min_values()`
`D`	A single threshold value
`Dt_max`	A logical value indicating whether the decision threshold is a maximum (`TRUE`) or a minimum (`FALSE`). The default is `TRUE`.
`risk_measures`	A list of risk scores generated by `calculate_risk()`. The policy alternatives in the `risk_measures` list must be in the same order as in the `max_min_values_list`.

Value

A list of ggplots, one for each policy alternative.

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
Dt <- rep(750, nrow(psa_data$Baseline))
D <- 750

risk_measures <- calculate_risk(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt = Dt,
  Dt_max = TRUE
)

peak_values_list <- get_max_min_values(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt_max = TRUE
)

density_plots <- plot_density(
  peak_values_list,
  D = D,
  Dt_max = TRUE,
  risk_measures
)
```

`plot_fan` *Generate time-outcome fan plots*

Description

For each policy alternative, this function generates time-outcome fan plots summarizing the trajectory of the outcome over time using the mean value for a given policy alternative. Uncertainty is characterized by shading the 50% and 95% credible intervals (calculated as 25th and 75th percentiles and 2.5th and 97.5th percentiles, respectively). The decision threshold is shown directly on the plot to provide a clear reference point for interpreting the outcome values.

Usage

```
plot_fan(data, tmin, tmax, Dt, Dt_max = TRUE)
```

Arguments

`data`	A list of data.frames (one data.frame for each policy alternative).
`tmin`	A numeric value or a date specifying the minimum simulation time to include in the analysis (ex. 1 or "2021-01-01"). This should correspond to a value in the first column of each data.frame in `data`.

tmax	A numeric value or a date specifying the maximum simulation time to include in the analysis (ex. 100 or "2021-04-10"). This should correspond to a value in the first column of each data.frame in data.
Dt	A numeric vector of decision thresholds, one for each model time step between tmin and tmax.
Dt_max	A logical value indicating whether the decision threshold is a maximum (TRUE) or a minimum (FALSE). The default is TRUE.

Value

A list of ggplots, one for each policy alternative.

data format

Each data.frame in data contains the results from multiple model runs using different parameter sets (e.g., from probabilistic sensitivity, uncertainty, or Bayesian inference analysis). The first column contains the model time and subsequent columns contain the predicted output for each simulation run at the respective time. The model time in the first column must contain numeric values indicating a simulation time (ex. 1, 2, 3,...) or dates (ex. "2021-01-01", "2021-01-02") which must be in R Date format (i.e., class="Date"). To ensure a consistent basis for comparison, the model time in the first column should be the same for each policy alternative (data.frame).

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
Dt <- rep(750, nrow(psa_data$Baseline))

fan_plots <- plot_fan(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt = Dt,
  Dt_max = TRUE
)
```

---

plot_raincloud *Generate raincloud plot*

Description

The probability densities of the highest (or lowest if the threshold is a minimum) projected outcome across simulation runs are plotted for each policy alternative alongside corresponding box plots, which indicate the mean and 50th percentile range. These plots are presented collectively on a single graph to facilitate visual comparison of the policy alternatives. The decision threshold is shown directly on the plot as a vertical line to provide a clear reference point for interpreting the outputs.

Usage

```
plot_raincloud(max_min_values_list, D)
```

Arguments

max_min_values_list	A list generated by get_max_min_values() that must contain at least two elements.
D	A single threshold value.

Value

A raincloud plot for peak (or minimum) values

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
D <- 750

peak_values_list <- get_max_min_values(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt_max = TRUE
)

plot_raincloud(
  peak_values_list,
  D = D
)
```

---

plot_temporal *Generate temporal probability density plots*

Description

The probability density of the peak (or lowest if the threshold is a minimum) projected outcome across simulation runs is plotted in the center of the graph for a given policy alternative. Above and below, the probability density of the outcome at specified time points relative to the time of the peak (or lowest) project outcome is plotted to visually illustrate how uncertainty, and therefore risk, changes over time. The decision threshold is shown directly on the plot as a vertical line to provide a clear reference point for interpreting the outputs.

Usage

```
plot_temporal(relative_values, D)
```

Arguments

relative_values	A list generated by get_relative_values().
D	A single threshold value.

Value

A list of ggplots, one for each policy alternative.

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
D <- 750
t_s <- 30
t_ss <- 10

peak_values_list <- get_max_min_values(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt_max = TRUE
)

peak_temporal_list <- get_relative_values(
  psa_data,
  peak_values_list,
  t_s = t_s,
  t_ss = t_ss
)

peak_temporal_plots <- plot_temporal(
  peak_temporal_list,
  D
)
```

---

psa_data *Sample PSA data*

Description

Sample simulation outputs for a baseline scenario and three interventions, in a format compatible with package functions

Usage

```
psa_data
```

Format

psa_data:

A list of data frames, one for each scenario corresponding to the list item name (Baseline, Intervention_1). Each data frame has 100 rows and 814 columns:

date (Baseline only) the simulation date in calendar time

time (Intervention scenarios only) simulation time as integer starting with 0

X1, ..., X1727 individual simulation results

`sum_stats_temporal` *Calculates summary statistics at specified time points relative to the peak (or minimum)*

Description

For each policy alternative, this function calculates summary statistics (n, mean, median, and IQR) for the model output values at the specified time points relative to the peak (or minimum) value.

Usage

```
sum_stats_temporal(relative_values)
```

Arguments

`relative_values`
: A list generated by `get_relative_values()`.

Value

A list of data.frames of summary statistics.

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
D <- 750
t_s <- 20
t_ss <- 10

peak_values_list <- get_max_min_values(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt_max = TRUE
)

peak_temporal_list <- get_relative_values(
  psa_data,
  peak_values_list,
```

```
  t_s = t_s,
  t_ss = t_ss
)

stats_peak_temporal <- sum_stats_temporal(peak_temporal_list)
```

`tabulate_risk` *Output risk measures to table*

Description

This function tabulates the expected risk values for each policy alternative and the percent change in risk relative to the baseline scenario.

Usage

```
tabulate_risk(exp_risk_list, n_s)
```

Arguments

`exp_risk_list`	A list of expected risk values where the first element corresponds to the baseline policy. This list can be generated using the `calculate_risk()` function.
`n_s`	A numeric value of the number of policy alternatives (including the baseline policy) to include in the table.

Value

A character matrix of risk values and policy risk impact (%) for each policy alternative.

Examples

```
tmin <- "2021-01-01"
tmax <- "2021-04-10"
Dt <- rep(750, nrow(psa_data$Baseline))

risk_measures <- calculate_risk(
  psa_data,
  tmin = tmin,
  tmax = tmax,
  Dt = Dt,
  Dt_max = TRUE
)

tabulate_risk(
  risk_measures,
  n_s = length(psa_data)
)
```

Index