Package: micromort (via r-universe)

June 8, 2026

Title Curated Micromort and Microlife Risk Datasets

Version 0.2.0

Description A data package providing curated, cross-sectional snapshots of micromort (acute risk) and microlife (chronic risk) values from authoritative sources including Wikipedia, CDC MMWR, and academic literature. Designed as a reference dataset for risk comparison and communication, not a time-series analysis tool or epidemiological modelling framework. Includes tools for risk comparison, lifestyle analysis, and a Plumber API for data distribution. All data normalized to unified schemas with full provenance tracking.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports arrow, checkmate, cli, dplyr, ggplot2, scales, tibble, units

Suggests box, bslib, digest, DT, eurostat, fs, gert, gh, here, htmltools, httr2, stringr, tidyr, jsonlite, knitr, pkgdown, plotly, pkgload, plumber, readr, rlang, rmarkdown, rprojroot, shiny, shinylive, targets, quarto, testthat (>= 3.0.0), urlchecker, usethis, yaml

Config/testthat/edition 3

VignetteBuilder quarto, knitr

URL `https://johngavin.github.io/micromort/`, `https://github.com/johngavin/micromort`

BugReports `https://github.com/johngavin/micromort/issues`

Config/pak/sysreqs cmake libssl-dev libudunits2-dev

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

Date/Publication 2026-06-08 10:30:48 UTC

RemoteUrl https://github.com/JohnGavin/micromort

RemoteRef HEAD

RemoteSha b14e0c7a59afbb48ea620b19d5a5cbb34efc5f4f

Contents

`activity_descriptions`

Activity Descriptions and Help URLs

Description

Returns a tibble of human-readable descriptions and authoritative help URLs for all activities in `common_risks()`. Useful for quiz tooltips, API responses, and dashboards.

Usage

```
activity_descriptions()
```

Value

A tibble with columns:

- `activity`: Activity name (matches `common_risks()` exactly)
- `description`: 1-2 sentence explanation of the risk
- `help_url`: Authoritative source URL (Wikipedia or similar)

Examples

```
activity_descriptions()
activity_descriptions() |> dplyr::filter(grepl("Skydiving", activity))
```

`acute_risks` *Acute Risks Dataset*

Description

A curated dataset of acute mortality risks measured in micromorts. One micromort equals a one-in-a-million chance of death.

Format

A tibble with 62 rows and 15 columns:

record_id Unique record identifier (source_id + sequence)

activity Human-readable activity name

activity_normalized Standardized activity name for grouping

micromorts Risk in micromorts (1 = one-in-a-million death risk)

microlives Equivalent in microlives (micromorts × 0.7)

category Activity category (Sport, Travel, Medical, etc.)

period Time period for risk (per event, per day, per year)

period_normalized Standardized period (event, day, week, month, year)

age_group Applicable age group (all, 18-49, 65+, etc.)

geography Geographic scope (global, US, UK, etc.)

year Year of data collection

source_id Source identifier (foreign key to risk_sources)

source_url Direct URL to source

confidence Data quality level (high, medium, low)

last_accessed Date data was retrieved

Details

Data is compiled from multiple sources including Wikipedia, micromorts.rip, and CDC MMWR reports. Multiple estimates for the same activity may exist from different sources.

Source

- Wikipedia: `https://en.wikipedia.org/wiki/Micromort`
- micromorts.rip: `https://micromorts.rip/`
- CDC MMWR: `https://www.cdc.gov/mmwr/`

References

Howard RA (1980). "On Making Life and Death Decisions." In Schwing & Albers (eds), Societal Risk Assessment.

See Also

Other datasets: `chronic_risks()`, `risk_sources`

Examples

```
# Load the acute risks dataset
acute <- load_acute_risks()
head(acute)

# Filter by category
acute |> dplyr::filter(category == "Sport")

# Top 10 riskiest activities
acute |> dplyr::slice_max(micromorts, n = 10)
```

`annual_risk_budget` *Annual Risk Budget*

Description

Calculate total annual micromort exposure from a list of activities.

Usage

```
annual_risk_budget(activities, age = NULL)
```

Arguments

`activities`	Named numeric vector of activity frequencies per year. Names should match activity names in `load_acute_risks()`.
`age`	Optional age for baseline risk calculation

Value

A tibble with risk budget breakdown including:

activity Activity name

frequency Times per year

micromorts_per Micromorts per occurrence

annual_micromorts Total annual micromorts

pct_of_total Percentage of total risk budget

See Also

`daily_hazard_rate()`, `load_acute_risks()`

Other analysis: `compare_interventions()`, `daily_hazard_rate()`, `lifestyle_tradeoff()`, `risk_sensitivity()`, `toxicological_risk()`

Examples

```
# Calculate annual risk from recreational activities
annual_risk_budget(c(
  "Skydiving (US)" = 10,
  "Scuba diving, trained" = 20,
  "Running a marathon" = 2
), age = 35)
```

`as_microlife` *Convert Minutes to Microlives*

Description

A microlife represents a 30-minute change in life expectancy per day. This function converts minutes of life expectancy change to microlives.

Usage

```
as_microlife(minutes, use_units = FALSE)
```

Arguments

`minutes`	Numeric. Life expectancy change in minutes. • Positive values = life gained (e.g., from exercise) • Negative values = life lost (e.g., from smoking)
`use_units`	Logical. If `TRUE`, returns a `units` object with unit ”microlife”. Default `FALSE` returns a plain numeric for backwards compatibility.

Details

Unit definition: 1 microlife = 30 minutes of life expectancy change per day.

Sign convention:

- Negative microlives = life expectancy loss (harmful)
- Positive microlives = life expectancy gain (beneficial)

Value

Numeric. Value in microlives (same sign as input), or a `units` object when `use_units = TRUE`.

See Also

`as_micromort()`, `lle()`, `chronic_risks()`

Other conversion: `as_micromort()`, `as_probability()`, `lle()`, `value_of_micromort()`

Examples

```
# Smoking 20 cigarettes/day: each costs ~30 mins = -600 mins total
as_microlife(-20 * 30)  # -20 microlives (life lost)

# Exercise 20 mins/day: gains ~60 mins life expectancy
as_microlife(60)        # +2 microlives (life gained)

# Being 5kg overweight: costs ~30 mins/day
as_microlife(-30)       # -1 microlife (life lost)

# With units tracking
as_microlife(60, use_units = TRUE) # 2 [microlife]
```

`as_micromort` *Convert Probability to Micromorts*

Description

A micromort represents a one-in-a-million chance of death. This function converts a raw probability of death into micromorts.

Usage

```
as_micromort(prob, use_units = FALSE)
```

Arguments

`prob`	Numeric. Probability of death (0 to 1).
`use_units`	Logical. If `TRUE`, returns a `units` object with unit "micromort". Default `FALSE` returns a plain numeric for backwards compatibility.

Value

Numeric value in micromorts, or a `units` object when `use_units = TRUE`.

See Also

`as_probability()`, `as_microlife()`, `value_of_micromort()`

Other conversion: `as_microlife()`, `as_probability()`, `lle()`, `value_of_micromort()`

Examples

```
as_micromort(1/1000000) # 1 micromort (plain numeric)
as_micromort(1/10000)   # 100 micromorts (plain numeric)
as_micromort(1/1000000, use_units = TRUE) # 1 [micromort]
```

`as_probability` *Convert Micromorts to Probability*

Description

Convert Micromorts to Probability

Usage

```
as_probability(micromorts)
```

Arguments

`micromorts`	Numeric. Risk in micromorts.

Value

Numeric probability.

See Also

`as_micromort()`, `as_microlife()`

Other conversion: `as_microlife()`, `as_micromort()`, `lle()`, `value_of_micromort()`

Examples

```
as_probability(1) # 1e-6
```

`atomic_risks` *Atomic Risk Components*

Description

Returns a tibble where each row represents ONE risk component of ONE activity. Different risk types (physical, medical, radiation) are never mixed in the same row. This is the foundational dataset from which `common_risks()` aggregates composite values.

Usage

```
atomic_risks()
```

Details

Activities that have not yet been decomposed use `component = "all_causes"` and `risk_category = "mixed"` as honest placeholders.

Value

A tibble with columns:

component_id Unique identifier: `{activity_id}_{component}_{condition}`

activity_id Groups components into one activity

activity Human-readable activity name with duration

component Risk component: `"all_causes"`, `"crash"`, `"dvt"`, `"radiation"`, etc.

risk_category `"physical"`, `"medical"`, `"radiation"`, `"environmental"`, `"mixed"`

component_label Human-readable label for this component

micromorts Risk for this component at this duration for this condition

duration_hours Activity duration this row applies to (`NA` for non-duration-dependent)

category Activity category: `"Travel"`, `"Medical"`, `"Daily Life"`, etc.

period Human-readable period: `"per day"`, `"per event"`, etc.

period_type `"event"`, `"day"`, `"hour"`, `"year"`, `"month"`, `"period"`

hedgeable Can this component be mitigated?

hedge_description How to mitigate (if hedgeable)

hedge_reduction_pct Estimated percent reduction from hedging

condition_variable What this risk depends on: `"health_profile"`, `"geography"`, `"country"`, `"age"`, or `NA`

condition_value Condition value: `"healthy"`, `"dvt_risk_factors"`, `"high_income"`, `"low_income"`, `"allergic"`, `"all_ages"`, age groups (e.g. `"under_65"`, `"65_74_male"`, `"85_plus_male"`), ISO-2 country codes (e.g. `"US"`, `"UK"`), or `NA`

confidence Data confidence: `"high"`, `"medium"`, `"low"`, `"estimated"`

source_url Citation URL

notes Scaling behavior, caveats

validation_status `"single_source"`, `"corroborated"`, or `"cross_validated"`

source_count Integer count of independent sources checked

estimate_range Character range (e.g. `"0.05-0.15"`) or `NA` for point estimates

See Also

`common_risks()` for the aggregated view.

Examples

```
atomic_risks()
atomic_risks() |> dplyr::filter(component != "all_causes")
atomic_risks() |> dplyr::filter(hedgeable)
```

`cancer_risks` *Cancer Risks by Type, Sex, and Age*

Description

Mortality rates for major cancers stratified by sex and age group, expressed in micromorts per year and daily microlives.

Usage

```
cancer_risks()
```

Details

Data from SEER Cancer Statistics (NCI) and American Cancer Society 2024-2026.

Value

A tibble with columns: cancer_type, sex, age_group, deaths_per_100k, micromorts_per_year, microlives_per_day, family_history_rr, rank_by_sex, source_url.

References

SEER Cancer Statistics Factsheets. National Cancer Institute. `https://seer.cancer.gov/statfacts/`

Siegel RL, et al. Cancer statistics, 2024. CA Cancer J Clin. 2024;74:12-49.

See Also

`vaccination_risks()`, `conditional_risk()`, `hedged_portfolio()`

Other conditional-risk: `conditional_risk()`, `hedged_portfolio()`, `vaccination_risks()`

Examples

```
cancer_risks()
cancer_risks() |> dplyr::filter(sex == "Female")
cancer_risks() |> dplyr::filter(age_group == "50-64")
```

`chronic_disease_risks` *Chronic Disease Risks by Country and Year*

Description

Returns daily micromorts from chronic diseases for one or more countries, using a bundled snapshot of IHME Global Burden of Disease data sourced via Our World in Data.

Usage

```
chronic_disease_risks(country = "GBR", year = NULL)
```

Arguments

`country`	Character vector of ISO-3 country codes (e.g. `"GBR"`, `c("GBR", "USA")`). Default is `"GBR"`. Use `"all"` to return every country in the bundled dataset.
`year`	Integer or `NULL`. If `NULL` (default), the most recent available year is returned. If a specific year is supplied, only rows matching that year are returned; an error is raised if the year is absent.

Details

The bundled CSV (`inst/extdata/owid_chronic_deaths.csv`) covers seven chronic cause categories across ~20 countries. To refresh the snapshot from the live OWID catalog, run `data-raw/owid_chronic_deaths.R`.

Conversion formula:

`daily_micromorts = (deaths_per_100k / 365) * 10`

Because: annual deaths per 100,000 → divide by 100,000 for annual probability → divide by 365 for daily probability → multiply by 1,000,000 for micromorts. The 100,000 and 1,000,000 cancel to a factor of 10: `rate / 100,000 / 365 × 1,000,000 = rate / 365 × 10`.

Value

A `tibble::tibble()` with columns:

cause Disease cause label (character)

country Country name (character)

iso3 ISO-3 country code (character)

year Data year (integer)

deaths_per_100k Age-standardised deaths per 100,000 per year (double)

daily_micromorts Daily micromort risk (double)

annual_micromorts Annual micromort risk (double)

References

Institute for Health Metrics and Evaluation (IHME). Global Burden of Disease Study 2019. Seattle, WA: IHME, 2020. https://www.healthdata.org/research-analysis/gbd

Our World in Data. Cause of Death. https://ourworldindata.org/causes-of-death

See Also

`chronic_risks()` for microlife-based chronic lifestyle factors.

Examples

```
# Default: UK, latest year
chronic_disease_risks()

# Specific country
chronic_disease_risks("USA")

# Multiple countries
chronic_disease_risks(c("GBR", "USA", "JPN"))

# All countries in bundled dataset
chronic_disease_risks("all")

# Filter by cause after calling
chronic_disease_risks("GBR") |>
  dplyr::filter(cause == "Cardiovascular diseases")
```

`chronic_quiz_pairs` *Generate Quiz Pairs for "Which Daily Habit Has a Bigger Effect?" Game*

Description

Creates candidate question pairs from `chronic_risks()` for use in an interactive microlife comparison quiz. Each pair contains two chronic lifestyle factors, and the player guesses which has the larger absolute effect on life expectancy (in microlives per day).

Usage

```
chronic_quiz_pairs(
  min_ratio = 1.1,
  max_ratio = 2,
  prefer_cross_category = TRUE,
  seed = NULL,
  difficulty = NULL
)
```

Arguments

`min_ratio`	Minimum ratio between absolute microlife values in a pair. Default 1.1. Ignored when `difficulty` is non-NULL.
`max_ratio`	Maximum ratio. Default 2.0. Ignored when `difficulty` is non-NULL.
`prefer_cross_category`	If `TRUE` (default), pairs from different categories are prioritised.
`seed`	Optional random seed for reproducibility.
`difficulty`	Optional difficulty level. One of `"easy"`, `"medium"`, `"hard"`, or `"mixed"`. When non-NULL, overrides `min_ratio`/`max_ratio` to use the full pool (ratios 1.5–10) and assigns difficulty via data-driven terciles.

Value

A tibble with columns:

- `factor_a`, `microlives_a`, `direction_a`, `category_a`, `description_a`, `help_url_a`, `annual_days_a`
- `factor_b`, `microlives_b`, `direction_b`, `category_b`, `description_b`, `help_url_b`, `annual_days_b`
- `ratio` (max/min of absolute microlife values)
- `answer` (”a” or ”b” – whichever factor has the larger absolute effect)
- `difficulty` (only when `difficulty` is non-NULL)

Examples

```
pairs <- chronic_quiz_pairs(seed = 42)
head(pairs)

easy <- chronic_quiz_pairs(difficulty = "easy", seed = 42)
head(easy)
```

`chronic_risks` *Chronic Risks Dataset*

Description

A curated dataset of chronic lifestyle factors measured in microlives. One microlife equals 30 minutes of life expectancy gained or lost.

A dataset of chronic lifestyle factors and their impact on life expectancy, measured in microlives (30 minutes of life expectancy per day).

Usage

```
chronic_risks()
```

Format

A tibble with 22 rows and 12 columns:

record_id Unique record identifier

factor Human-readable factor name

factor_normalized Standardized factor name for grouping

microlives_per_day Daily impact in microlives (+/- 30 min units)

direction Effect direction: "gain" or "loss"

category Factor category (Diet, Exercise, Smoking, etc.)

description Detailed description of the factor

annual_effect_days Days of life gained/lost per year

source_id Source identifier

source_url Direct URL to source

confidence Data quality level

last_accessed Date data was retrieved

Details

Positive values indicate life expectancy gains; negative values indicate losses. Based on the framework introduced by David Spiegelhalter (2012).

Positive values indicate life expectancy gains; negative values indicate losses. Effects are cumulative over a lifetime of adult exposure (~57 years).

Value

A tibble with columns: factor, microlives_per_day, category, direction, annual_effect_days, source_url.

Source

Spiegelhalter D (2012). "Using speed of ageing and 'microlives' to communicate the effects of lifetime habits and environment." BMJ 2012;345:e8223. doi:10.1136/bmj.e8223

References

`https://en.wikipedia.org/wiki/Microlife`

Spiegelhalter D (2012). "Using speed of ageing and 'microlives' to communicate the effects of lifetime habits and environment." BMJ 2012;345:e8223. doi:10.1136/bmj.e8223

`https://en.wikipedia.org/wiki/Microlife`

`https://pubmed.ncbi.nlm.nih.gov/23247978/`

See Also

Other datasets: `acute_risks`, `risk_sources`

Examples

```
# Load the chronic risks dataset
chronic <- load_chronic_risks()
head(chronic)

# Factors that reduce life expectancy
chronic |> dplyr::filter(direction == "loss")

# Factors that increase life expectancy
chronic |> dplyr::filter(direction == "gain")
chronic_risks()
chronic_risks() |> dplyr::filter(direction == "loss")
chronic_risks() |> dplyr::filter(category == "Exercise")
```

`combined_quiz_pairs` *Generate cross-domain quiz pairs comparing acute and chronic risks*

Description

Creates candidate question pairs mixing acute risks from `common_risks()` (micromorts per event) with chronic risks from `chronic_risks()` (microlives per day), converting both to a common microlife scale so they can be directly compared. Each pair pits one acute activity against one chronic lifestyle factor and asks which has a larger total impact over `time_period_days`.

Usage

```
combined_quiz_pairs(n = 10, time_period_days = 365, seed = NULL)
```

Arguments

`n`	Integer. Number of pairs to generate. Default `10`.
`time_period_days`	Numeric. Time period for chronic risk accumulation. Default `365` (1 year). For chronic risks, the cumulative impact is `abs(microlives_per_day) × time_period_da`
`seed`	Integer or `NULL`. Random seed for reproducibility. Default `NULL` (random each call).

Details

The conversion from micromorts to microlives uses the factor `0.7` (1 micromort 0.7 microlives at age 40). Chronic impact over the time period is `abs(microlives_per_day) × time_period_days`.

Value

A tibble with columns:

- `activity_a`, `type_a` (”acute”), `value_a`, `unit_a` (”micromorts”), `category_a`, `period_a`
- `factor_b`, `type_b` (”chronic”), `value_b`, `unit_b` (”microlives/day”), `category_b`, `direction_b`
- `common_unit` (”microlives”), `common_value_a`, `common_value_b`
- `correct_answer` (”a” or ”b” — whichever has higher impact in common unit)
- `ratio` (larger / smaller common value)
- `explanation` describing both values and the conversion

Examples

```
pairs <- combined_quiz_pairs(n = 5, seed = 42)
pairs[, c("activity_a", "factor_b", "correct_answer", "ratio")]

# One year time period (default)
pairs_year <- combined_quiz_pairs(n = 10, time_period_days = 365, seed = 1)
```

`common_risks` *Acute Risks in Micromorts*

Description

A comprehensive dataset of activities and their associated acute mortality risk in micromorts, with calculated microlives and source references.

Usage

```
common_risks(profile = list(), duration_hours = NULL)
```

Arguments

`profile`	A named list of condition variables for filtering conditional risks, e.g. `list(health_profile = "dvt_risk_factors")` or `list(country = "US")` for country-specific road traffic and homicide risks. Default `list()` returns unconditional/healthy defaults.
`duration_hours`	Optional numeric. For duration-dependent activities, selects the nearest pre-computed duration bucket *within each activity*. All flying variants (2h, 5h, 8h, 12h) are retained. `NULL` (default) returns all duration buckets. Use `risk_for_duration()` to select a single activity family result.

Details

Aggregates from `atomic_risks()`, summing component-level micromorts per activity.

Micromort: one-in-a-million chance of death (acute risk). Microlife: 30 minutes of life expectancy lost.

Data sources: Wikipedia, micromorts.rip, CDC MMWR, academic literature.

Value

A tibble with columns:

activity Activity name

micromorts Risk in micromorts (1 = 1-in-a-million death probability)

microlives Equivalent microlives (micromorts x 0.7)

category Activity category

period Human-readable period description

period_type Normalized period type: "event", "day", "hour", "year", "period"

period_days Typical duration in days (for cross-activity comparison)

micromorts_per_day Micromorts normalized per day

source_url Data source URL

n_components Number of atomic components summed

hedgeable_pct Percent of total micromorts that are hedgeable

References

Howard RA (1980). "On Making Life and Death Decisions." In Schwing & Albers (eds), Societal Risk Assessment: How Safe Is Safe Enough?

`https://en.wikipedia.org/wiki/Micromort`

`https://micromorts.rip/`

See Also

`atomic_risks()` for the component-level data.

Examples

```
common_risks()
common_risks() |> dplyr::filter(category == "COVID-19")
common_risks() |> dplyr::filter(micromorts > 100)
```

`compare_interventions` *Compare Lifestyle Interventions*

Description

Compare the impact of multiple lifestyle changes in microlives. Uses the chronic risks dataset to calculate daily, annual, and lifetime effects of interventions.

Usage

```
compare_interventions(interventions)
```

Arguments

`interventions` Named list of interventions. Each element should have:

- **factor** Name of the chronic risk factor (must match `chronic_risks()`)
- **change** Numeric change multiplier (e.g., -1 to remove the factor)

Value

A tibble comparing effects of each intervention:

intervention Name of the intervention

factor Original factor name

original_ml_per_day Original microlives per day

change Change multiplier applied

net_ml_per_day Net microlives gained/lost per day

annual_days Days of life gained/lost per year

lifetime_years Years of life gained/lost over 57 years

See Also

`lifestyle_tradeoff()`, `daily_hazard_rate()`, `annual_risk_budget()`

Other analysis: `annual_risk_budget()`, `daily_hazard_rate()`, `lifestyle_tradeoff()`, `risk_sensitivity()`, `toxicological_risk()`

Examples

```
# Compare quitting smoking vs losing weight
compare_interventions(list(
  "Quit 10 cigarettes/day" = list(factor = "Smoking 10 cigarettes", change = -1),
  "Lose 5kg" = list(factor = "Being 5 kg overweight", change = -1)
))
```

`conditional_risk` *Conditional Risk Comparison (Hedged vs Unhedged)*

Description

Compare mortality risk between ”hedged” (optimal lifestyle/interventions) and ”unhedged” (baseline/suboptimal) scenarios for major disease categories.

Usage

```
conditional_risk(disease = "all")
```

Arguments

`disease`	Character. Disease category: ”cardiovascular”, ”cancer”, ”respiratory”, ”infectious”, or ”all”.

Value

A tibble comparing hedged vs unhedged risks in micromorts and microlives.

See Also

`cancer_risks()`, `vaccination_risks()`, `hedged_portfolio()`

Other conditional-risk: `cancer_risks()`, `hedged_portfolio()`, `vaccination_risks()`

Examples

```
conditional_risk("cardiovascular")
conditional_risk("cancer")
conditional_risk("all")
```

`covid_vaccine_rr` *COVID-19 Vaccine Relative Risks*

Description

Mortality risk comparison between vaccinated and unvaccinated populations during the Omicron BA.4/BA.5 period (Sep-Dec 2022).

Usage

```
covid_vaccine_rr()
```

Details

Data source: CDC MMWR Vol. 72, No. 6 (Feb 2023).

Value

A tibble with vaccination status, death rates, micromorts, microlives, and relative risk compared to bivalent booster recipients.

References

Link SC, et al. COVID-19 Incidence and Death Rates Among Unvaccinated and Fully Vaccinated Adults with and Without Booster Doses During Periods of Delta and Omicron Variant Emergence. MMWR 2023;72:132-138. `https://www.cdc.gov/mmwr/volumes/72/wr/mm7206a3.htm`

Examples

```
covid_vaccine_rr()
covid_vaccine_rr() |> dplyr::filter(age_group == "All ages")
```

`daily_hazard_rate` *Daily Hazard Rate by Age*

Description

Calculates the daily probability of death based on age using a simplified Gompertz-Makeham mortality model. Returns a central estimate plus credible bounds derived from ±10% sensitivity on the Gompertz parameters `a` (background mortality) and `b` (initial mortality).

Usage

```
daily_hazard_rate(age, sex = "male")
```

Arguments

`age`	Age in years
`sex`	”male” or ”female” (default: ”male”)

Details

The Gompertz hazard is `h(x) = a + b * exp(c * x)`. Bounds are computed by varying `a` and `b` independently by ±10% (four combinations) and taking the min/max across those combinations.

Value

A tibble with:

age Input age

sex Input sex

daily_prob Daily probability of death (central estimate)

micromorts Daily baseline risk in micromorts (central estimate)

micromorts_lower Lower credible bound (micromorts) from ±10% parameter sensitivity

micromorts_upper Upper credible bound (micromorts) from ±10% parameter sensitivity

microlives_consumed Estimated microlives consumed per day

precision_note Reminder that Gompertz parameters are approximate

interpretation Human-readable summary

References

Gompertz B (1825). "On the Nature of the Function Expressive of the Law of Human Mortality." Philosophical Transactions of the Royal Society.

See Also

`annual_risk_budget()`, `compare_interventions()`

Other analysis: `annual_risk_budget()`, `compare_interventions()`, `lifestyle_tradeoff()`, `risk_sensitivity()`, `toxicological_risk()`

Examples

```
# Baseline risk at age 30 with credible bounds
daily_hazard_rate(30)

# Compare male vs female at age 65
daily_hazard_rate(65, "male")
daily_hazard_rate(65, "female")
```

`demographic_factors` *Demographic Life Expectancy Factors*

Description

Population-level factors affecting baseline life expectancy, expressed as microlives per day relative to a reference population.

Usage

```
demographic_factors()
```

Details

Based on actuarial data and epidemiological studies.

Value

A tibble with demographic factors and their microlife effects.

References

Spiegelhalter D (2012). BMJ 2012;345:e8223.

https://en.wikipedia.org/wiki/Microlife

Examples

```
demographic_factors()
```

`export_combined_quiz_csv`

Export combined quiz pairs to CSV for Shinylive

Description

Generates a representative set of combined quiz pairs and writes them to `inst/extdata/combined_quiz_pairs.csv` The Shinylive combined quiz can read this CSV directly without requiring R computation in the browser.

Usage

```
export_combined_quiz_csv(
  n = 50L,
  time_period_days = 365,
  seed = 42L,
  path = NULL
)
```

Arguments

`n`	Integer. Number of pairs to export. Default `50`.
`time_period_days`	Numeric. Time period for chronic risk accumulation. Default `365` (1 year).
`seed`	Integer. Random seed for reproducibility. Default `42`.
`path`	Character. Output file path. Default writes to `inst/extdata/combined_quiz_pairs.csv` under the package root.

Value

Path to the written CSV (invisibly).

Examples

```
## Not run:
export_combined_quiz_csv()

## End(Not run)
```

`factor_descriptions` *Chronic Factor Descriptions and Help URLs*

Description

Returns a tibble of human-readable descriptions and authoritative help URLs for all factors in `chronic_risks()`. Useful for quiz tooltips, API responses, and dashboards.

Usage

```
factor_descriptions()
```

Value

A tibble with columns:

- `factor`: Factor name (matches `chronic_risks()` exactly)
- `description`: 1-2 sentence explanation of the factor
- `help_url`: Authoritative source URL

Examples

```
factor_descriptions()
factor_descriptions() |> dplyr::filter(grepl("exercise", factor, ignore.case = TRUE))
```

`format_activity_name` *Format Activity Name with Line Break Before Parenthetical*

Description

Inserts an HTML `<br>` before the first opening parenthesis in an activity name, making quiz buttons more readable by separating the qualifier.

Usage

```
format_activity_name(name)
```

Arguments

`name`	Character string. The activity name to format.

Value

A `shiny::HTML()` object with `<br>` inserted before `(`, or the original string wrapped in `HTML()` if no parenthesis is present.

Examples

```
format_activity_name("airline pilot (annual radiation)")
format_activity_name("Skydiving")
```

`geography_quiz_pairs` *Generate Geography Quiz Pairs Comparing Risk Across Countries*

Description

Creates quiz pairs that compare the same disease risk across different countries, or compare country-specific disease risk against acute one-off activities. Uses `common_risks()` with country profiles.

Usage

```
geography_quiz_pairs(
  countries = c("UK", "NG"),
  seed = NULL,
  include_acute = TRUE,
  difficulty = NULL
)
```

Arguments

`countries`	Character vector of ISO-2 country codes to compare. Default `c("UK", "NG")` (UK vs Nigeria — high contrast).
`seed`	Optional random seed for reproducibility.
`include_acute`	If `TRUE` (default), also includes cross-domain pairs comparing country-specific disease risk vs unconditional acute activities (e.g., ”Daily CVD risk (Nigeria) vs Skydiving”).
`difficulty`	Optional difficulty level: `"easy"`, `"medium"`, `"hard"`, or `"mixed"`.

Value

A tibble with the same columns as `quiz_pairs()`, plus a `pair_type` column indicating `"cross_country"` or `"disease_vs_acute"`.

Examples

```
geography_quiz_pairs(countries = c("UK", "NG"), seed = 42)
```

`hedged_portfolio` *Hedged Portfolio Risk Summary*

Description

Calculate total risk reduction from adopting an optimal "hedged" lifestyle across multiple disease categories.

Usage

```
hedged_portfolio(
  include_diseases = c("cardiovascular", "cancer", "respiratory", "infectious")
)
```

Arguments

`include_diseases`	Character vector. Which disease categories to include. Default is all: cardiovascular, cancer, respiratory, infectious.

Value

A tibble with total hedged vs unhedged comparison and breakdown.

See Also

`cancer_risks()`, `vaccination_risks()`, `conditional_risk()`

Other conditional-risk: `cancer_risks()`, `conditional_risk()`, `vaccination_risks()`

Examples

```
hedged_portfolio()
hedged_portfolio(include_diseases = c("cardiovascular", "cancer"))
```

`infectious_disease_risks` *Daily micromorts from infectious diseases by country*

Description

Returns daily micromorts from infectious diseases for one or more countries, using a bundled snapshot of IHME Global Burden of Disease data sourced via Our World in Data.

Usage

```
infectious_disease_risks(country = "GBR", year = NULL)
```

Arguments

`country`	Character vector of ISO-3 country codes (e.g. `"GBR"`, `c("GBR", "USA")`). Default is `"GBR"`. Use `"all"` to return every country in the bundled dataset.
`year`	Integer or `NULL`. If `NULL` (default), the most recent available year is returned. If a specific year is supplied, only rows matching that year are returned; an error is raised if the year is absent.

Details

The bundled CSV (`inst/extdata/owid_infectious_deaths.csv`) covers seven infectious cause categories across 26 countries. To refresh the snapshot, run `data-raw/owid_infectious_deaths.R`.

Conversion formula:

`daily_micromorts = (deaths_per_100k / 365) * 10`

Because: annual deaths per 100,000 → divide by 100,000 for annual probability → divide by 365 for daily probability → multiply by 1,000,000 for micromorts. The 100,000 and 1,000,000 cancel to a factor of 10: `rate / 100,000 / 365 × 1,000,000 = rate / 365 × 10`.

Value

A `tibble::tibble()` with columns:

cause Disease cause label (character)

country Country name (character)

iso3 ISO-3 country code (character)

year Data year (integer)

deaths_per_100k Age-standardised deaths per 100,000 per year (double)

daily_micromorts Daily micromort risk (double)

annual_micromorts Annual micromort risk (double)

References

Institute for Health Metrics and Evaluation (IHME). Global Burden of Disease Study 2019. Seattle, WA: IHME, 2020. `https://www.healthdata.org/research-analysis/gbd`

Our World in Data. Cause of Death. `https://ourworldindata.org/causes-of-death`

See Also

`chronic_disease_risks()` for chronic disease micromort risks by country.

Examples

```
# Default: UK, latest year
infectious_disease_risks()

# Specific country
infectious_disease_risks("USA")
```

```
# Multiple countries
infectious_disease_risks(c("GBR", "USA", "IND"))

# All countries in bundled dataset
infectious_disease_risks("all")

# Filter by cause after calling
infectious_disease_risks("GBR") |>
  dplyr::filter(cause == "Tuberculosis")
```

`kendall_tau_score` *Kendall Tau Score for Ranking Quiz*

Description

Computes how well a user's ranking matches the correct ranking using the Kendall tau distance. Counts concordant pairs (user agrees with correct order) vs discordant pairs (user disagrees).

Usage

```
kendall_tau_score(user_order, correct_order)
```

Arguments

`user_order`	Integer vector of item indices in the user's order (first element = user's #1 pick, i.e. most risky).
`correct_order`	Integer vector of item indices in the correct order.

Value

A list with:

- `score`: number of concordant pairs
- `max_score`: total number of pairs = k*(k-1)/2
- `n_concordant`: same as score
- `n_discordant`: pairs where user and correct order disagree
- `pct`: percentage score (0-100)

Examples

```
# Perfect score
kendall_tau_score(c(1, 2, 3), c(1, 2, 3))

# Completely reversed
kendall_tau_score(c(3, 2, 1), c(1, 2, 3))
```

```
# One swap
kendall_tau_score(c(2, 1, 3), c(1, 2, 3))
```

`laggard_regions` *Laggard Regions with Stalled Life Expectancy Gains*

Description

Convenience function returning regions classified as "laggard" - those with lower life expectancy or stagnant improvement trends since 2005.

Usage

```
laggard_regions(country = NULL, year = NULL, sex = NULL)
```

Arguments

`country`	Character vector. Filter to specific countries using ISO 2-letter codes (e.g., "FR", "DE", "ES"). Default `NULL` returns all countries.
`year`	Integer or vector. Filter to specific years. Default `NULL` returns all years (1992-2023).
`sex`	Character. Filter by sex: "Male", "Female", or "Total". Default `NULL` returns all.

Value

A tibble filtered to laggard regions only.

See Also

`regional_life_expectancy()`, `vanguard_regions()`

Other regional: `regional_life_expectancy()`, `regional_mortality_multiplier()`, `vanguard_regions()`

Examples

```
# Laggard regions in 2019
laggard_regions(year = 2019, sex = "Total")
```

`launch_api` *Launch Micromort REST API*

Description

Starts a Plumber API server exposing 27 endpoints for accessing micromort and microlife risk datasets. Every response uses a standard JSON envelope with `data` and `meta` fields including source provenance.

Usage

```
launch_api(host = "127.0.0.1", port = 8080, docs = TRUE)
```

Arguments

`host`	Host to bind to (default: ”127.0.0.1”)
`port`	Port to listen on (default: 8080)
`docs`	Enable Swagger documentation (default: TRUE)

Value

Invisible NULL. Runs the API server until interrupted.

Core Risks (8 GET)

- `GET /v1/risks/acute` — Enriched acute risks (common_risks)
- `GET /v1/risks/acute/atomic` — Atomic risk components
- `GET /v1/risks/chronic` — Chronic microlife gains/losses
- `GET /v1/risks/cancer` — Cancer risk by type/sex/age
- `GET /v1/risks/vaccination` — Vaccination risk reduction
- `GET /v1/risks/covid-vaccine` — COVID vaccine relative risks
- `GET /v1/risks/conditional` — Conditional risk given disease
- `GET /v1/risks/demographic` — Demographic risk factors

Regional (4 GET)

- `GET /v1/regional/life-expectancy` — Regional life expectancy
- `GET /v1/regional/vanguard` — Best-performing regions
- `GET /v1/regional/laggard` — Worst-performing regions
- `GET /v1/regional/mortality-multiplier` — Mortality multiplier

Radiation (2 GET)

- `GET /v1/radiation/profiles` — Exposure by career milestones
- `GET /v1/radiation/patient-comparison` — Patient vs occupational

Analysis (2 GET + 4 POST)

- `GET /v1/analysis/equivalence` — Risk equivalence lookup
- `GET /v1/analysis/tradeoff` — Lifestyle tradeoff calculator
- `POST /v1/analysis/exchange-matrix` — Risk exchange matrix
- `POST /v1/analysis/interventions` — Compare interventions
- `POST /v1/analysis/budget` — Annual risk budget
- `POST /v1/analysis/hedged-portfolio` — Hedged risk portfolio

Conversion (6 GET)

- `GET /v1/convert/to-micromort` — Probability to micromorts
- `GET /v1/convert/to-probability` — Micromorts to probability
- `GET /v1/convert/to-microlife` — Minutes to microlives
- `GET /v1/convert/value` — Monetary value of one micromort
- `GET /v1/convert/lle` — Loss of life expectancy
- `GET /v1/convert/hazard-rate` — Daily hazard rate by age

Quiz (1 GET)

- `GET /v1/quiz/pairs` — Quiz pairs for comparison game

Metadata (3 endpoints)

- `GET /v1/sources` — Risk data sources registry
- `GET /v1/meta` — API metadata and endpoint listing
- `GET /health` — Health check

Examples

```
## Not run:
launch_api()

# Example requests (from another terminal):
# curl http://localhost:8080/v1/risks/acute?category=Sport
# curl http://localhost:8080/v1/risks/chronic?direction=gain
# curl http://localhost:8080/v1/convert/hazard-rate?age=35

## End(Not run)
```

`launch_chronic_quiz` *Launch Interactive "Which Daily Habit Has a Bigger Effect?" Quiz*

Description

A standalone Shiny app where users compare pairs of chronic lifestyle factors and guess which has the larger absolute effect on life expectancy (microlives per day). Built with bslib cards for a modern UI.

Usage

```
launch_chronic_quiz(n_pairs = NULL, ...)
```

Arguments

`n_pairs`	Number of question pairs. If `NULL` (default), the user chooses on the instructions page.
`...`	Additional arguments passed to `shiny::shinyApp()`.

Value

A Shiny app object (runs interactively).

Examples

```
if (interactive()) {
  launch_chronic_quiz()
}
```

`launch_dashboard` *Launch Risk Explorer Dashboard*

Description

Starts an interactive Shiny dashboard for exploring micromort and microlife data.

Usage

```
launch_dashboard(...)
```

Arguments

`...`	Additional arguments passed to shiny::runApp()

Value

Invisible NULL. Runs the dashboard until closed.

Examples

```
## Not run:
launch_dashboard()

## End(Not run)
```

`launch_quiz` *Launch Interactive "Which Is Riskier?" Quiz*

Description

A standalone Shiny app where users compare pairs of risky activities and guess which carries more micromort risk. Built with bslib cards for a modern UI.

Usage

```
launch_quiz(n_pairs = NULL, ...)
```

Arguments

`n_pairs`	Number of question pairs to offer as options (5 or 10). If `NULL` (default), the user chooses on the instructions page.
`...`	Additional arguments passed to `shiny::shinyApp()`.

Value

A Shiny app object (runs interactively).

Examples

```
if (interactive()) {
  launch_quiz()
}
```

`lifestyle_tradeoff` *Calculate Lifestyle Tradeoff*

Description

Calculate how much of one good habit compensates for a bad habit.

Usage

```
lifestyle_tradeoff(bad_habit, good_habit)
```

Arguments

`bad_habit`	Factor name of the bad habit (from `chronic_risks()`)
`good_habit`	Factor name of the compensating behavior (from `chronic_risks()`)

Value

A tibble showing the tradeoff ratio

See Also

`compare_interventions()`, `chronic_risks()`

Other analysis: `annual_risk_budget()`, `compare_interventions()`, `daily_hazard_rate()`, `risk_sensitivity()`, `toxicological_risk()`

Examples

```
# How much exercise offsets 2 cigarettes?
lifestyle_tradeoff("Smoking 2 cigarettes", "20 min moderate exercise")
```

`lle` *Loss of Life Expectancy (LLE)*

Description

Estimates the average time lost from a lifespan due to a specific risk.

Usage

```
lle(prob, life_expectancy = 40)
```

Arguments

`prob`	Numeric. Probability of death.
`life_expectancy`	Numeric. Remaining life expectancy in years (default 40).

Value

Numeric. Loss of life expectancy in seconds, minutes, or days (estimated).

See Also

`as_micromort()`, `as_microlife()`, `value_of_micromort()`

Other conversion: `as_microlife()`, `as_micromort()`, `as_probability()`, `value_of_micromort()`

Examples

```
lle(1/1000000, 40) # Loss from 1 micromort
```

`load_acute_risks` *Load Acute Risks Dataset*

Description

Loads the acute risks parquet dataset from inst/extdata/.

Usage

```
load_acute_risks()
```

Value

A tibble with acute risk data

Examples

```
acute <- load_acute_risks()
nrow(acute)
```

`load_chronic_risks` *Load Chronic Risks Dataset*

Description

Loads the chronic risks parquet dataset from inst/extdata/.

Usage

```
load_chronic_risks()
```

Value

A tibble with chronic risk data

Examples

```
chronic <- load_chronic_risks()
nrow(chronic)
```

`load_sources` *Load Risk Sources Registry*

Description

Loads the risk sources parquet dataset from inst/extdata/.

Usage

```
load_sources()
```

Value

A tibble with source metadata

Examples

```
sources <- load_sources()
nrow(sources)
```

`patient_radiation_comparison` *Patient vs Occupational Radiation Comparison*

Description

Cross-tabulates patient X-ray exposure against occupational career radiation to reveal surprising equivalences. For example, 100 lifetime chest X-rays (10 micromorts) exceeds a 40-year X-ray technician career (2 micromorts).

Usage

```
patient_radiation_comparison(
  xray_counts = c(1, 10, 100),
  career_years = c(10, 20, 40)
)
```

Arguments

`xray_counts`	Integer vector of patient X-ray counts to compare. Default `c(1, 10, 100)`.
`career_years`	Integer vector of occupational career durations. Default `c(10, 20, 40)`.

Value

A tibble with columns: occupation, xray_count, career_years, patient_micromorts, occupational_micromorts, ratio.

See Also

`radiation_profiles()`, `atomic_risks()`

Examples

```
patient_radiation_comparison()
patient_radiation_comparison(xray_counts = c(50, 200), career_years = c(5, 30))
```

`plot_risk_components` *Plot Risk Components as Stacked Bar*

Description

Creates a stacked bar chart showing the breakdown of atomic risk components for selected activities. Hedgeable components are visually distinguished.

Usage

```
plot_risk_components(activity_ids, profile = list(), risks = NULL)
```

Arguments

`activity_ids`	Character vector of activity IDs to plot.
`profile`	A named list of condition variables for filtering.
`risks`	Optional pre-computed `atomic_risks()` tibble.

Value

A ggplot2 object.

See Also

`risk_components()`, `atomic_risks()`

Examples

```
plot_risk_components(c("flying_2h", "flying_8h", "flying_12h"))
```

`plot_risks` *Plot Risk Comparison*

Description

Visualizes the risk of different activities in micromorts. For filtering by category, use `prepare_risks_plot()` first.

Usage

```
plot_risks(
  risks = common_risks(),
  facet = TRUE,
  height = 12,
  label_size = 9,
  dark = TRUE,
  guide_lines = TRUE,
  jitter_ones = TRUE,
  cluster_bands = TRUE
)
```

Arguments

`risks`	Tibble. Dataframe of risks from `prepare_risks_plot()` or `common_risks()`. If not pre-filtered, applies default filtering.
`facet`	Logical. If TRUE, splits plot into COVID-19 and Other panels. Default is TRUE.
`height`	Numeric. Plot height in inches. Default is 12.
`label_size`	Numeric. Y-axis label font size. Default is 9.
`dark`	Logical. If TRUE (default), use `theme_micromort_dark()`. If FALSE, use `theme_minimal()`.
`guide_lines`	Logical. If TRUE (default), add dashed guide lines from y-axis labels to bar starts.
`jitter_ones`	Logical. If TRUE (default), shift 1-micromort values slightly so bars are visible on log scale.
`cluster_bands`	Logical. If TRUE (default), add subtle background bands grouping risks with similar micromort values.

Value

A ggplot2 object.

See Also

`prepare_risks_plot()`, `plot_risks_interactive()`, `common_risks()`

Other visualization: `plot_risks_interactive()`, `prepare_risks_plot()`, `theme_micromort_dark()`

Examples

```
# Default dark plot
plot_risks()

# Light theme
plot_risks(dark = FALSE)

# Filter then plot
prepare_risks_plot(categories = "Sport") |> plot_risks()

# Exclude COVID-19 and show top 20
prepare_risks_plot(exclude_categories = "COVID-19", top_n = 20) |>
  plot_risks(facet = FALSE)
```

`plot_risks_interactive`

Interactive Risk Plot

Description

Creates an interactive plotly visualization of risks with category filtering.

Usage

```
plot_risks_interactive(risks = common_risks())
```

Arguments

`risks`	Tibble. Dataframe of risks, defaults to `common_risks()`.

Value

A plotly object with interactive filtering.

See Also

`plot_risks()`, `common_risks()`

Other visualization: `plot_risks()`, `prepare_risks_plot()`, `theme_micromort_dark()`

Examples

```
if (requireNamespace("plotly", quietly = TRUE)) {
  plot_risks_interactive()
}
```

`prepare_risks_plot` *Prepare Risk Data for Plotting*

Description

Filters and prepares risk data for visualization. Use this to filter categories before passing to `plot_risks()`.

Usage

```
prepare_risks_plot(
  risks = common_risks(),
  categories = NULL,
  exclude_categories = NULL,
  min_micromorts = 0.1,
  top_n = NULL
)
```

Arguments

`risks`	Tibble. Dataframe of risks, defaults to `common_risks()`.
`categories`	Character vector. Categories to include. Use `NULL` (default) for all categories. See `common_risks()` for available categories.
`exclude_categories`	Character vector. Categories to exclude. Applied after `categories` filter.
`min_micromorts`	Numeric. Minimum micromorts to include (default 0.1 to avoid invisible bars on log scale).
`top_n`	Integer. If specified, return only the top N risks by micromorts.

Value

A tibble ready for plotting with `plot_risks()`.

See Also

`plot_risks()`, `common_risks()`

Other visualization: `plot_risks()`, `plot_risks_interactive()`, `theme_micromort_dark()`

Examples

```
# All risks
prepare_risks_plot()

# Only COVID-19 risks
prepare_risks_plot(categories = "COVID-19")
```

```
# Exclude COVID-19
prepare_risks_plot(exclude_categories = "COVID-19")

# Multiple categories
prepare_risks_plot(categories = c("Sport", "Travel"))

# Top 20 risks
prepare_risks_plot(top_n = 20)

# Chain with plotting
prepare_risks_plot(categories = "Sport") |> plot_risks()
```

`quiz_pairs` *Generate Quiz Pairs for "Which Is Riskier?" Game*

Description

Creates candidate question pairs from `common_risks()` for use in an interactive risk comparison quiz. Each pair contains two activities with similar micromort values, making the comparison challenging and educational.

Usage

```
quiz_pairs(
  min_ratio = 1.1,
  max_ratio = 2,
  prefer_cross_category = TRUE,
  seed = NULL,
  difficulty = NULL,
  profile = list()
)
```

Arguments

`min_ratio`	Minimum ratio between micromort values in a pair. Values above 1.0 exclude identical-risk pairs that are unanswerable. Default 1.1. Ignored when `difficulty` is non-NULL.
`max_ratio`	Maximum ratio between micromort values in a pair. Lower values produce harder questions. Default 2.0. Ignored when `difficulty` is non-NULL.
`prefer_cross_category`	If `TRUE` (default), pairs from different risk categories are prioritised over same-category pairs.
`seed`	Optional random seed for reproducibility.
`difficulty`	Optional difficulty level. One of `"easy"`, `"medium"`, `"hard"`, or `"mixed"`. When non-NULL, overrides `min_ratio` and `max_ratio` to use the full pool (ratios 1.5–10) and assigns difficulty based on data-driven terciles:

- **hard**: lowest third of ratios (hardest to distinguish)
- **medium**: middle third
- **easy**: highest third (easiest to distinguish)
- **mixed**: samples equally from all three tiers

When `NULL` (default), the original `min_ratio`/`max_ratio` behaviour is preserved and no `difficulty` column is added.

`profile`	A named list of condition variables for filtering conditional risks, passed to `common_risks()`. E.g. `list(country = "US")` to include country-specific disease mortality in the quiz pool. Default `list()` uses population-average unconditional risks only.

Value

A tibble with columns:

- `activity_a`, `micromorts_a`, `category_a`, `hedgeable_pct_a`, `period_a`
- `activity_b`, `micromorts_b`, `category_b`, `hedgeable_pct_b`, `period_b`
- `description_a`, `help_url_a`, `description_b`, `help_url_b`
- `ratio` (max/min of the two micromort values)
- `answer` (”a” or ”b” — whichever activity is riskier)
- `difficulty` (only when `difficulty` is non-NULL)

Examples

```
pairs <- quiz_pairs(seed = 42)
head(pairs)

# Easy questions (large ratio differences)
easy <- quiz_pairs(difficulty = "easy", seed = 42)
head(easy)

# Country-specific quiz with Nigerian disease risk
ng_pairs <- quiz_pairs(profile = list(country = "NG"), seed = 42)
head(ng_pairs)
```

`radiation_profiles` *Radiation Exposure Profiles*

Description

Compares annual and cumulative radiation exposure across occupational, passenger, and environmental profiles. Uses the Linear No-Threshold (LNT) model for dose-to-risk conversion.

Usage

```
radiation_profiles(milestones = c(10, 20, 40))
```

Arguments

`milestones`	Integer vector of career/exposure years for cumulative columns. Default `c(10, 20, 40)`.

Value

A tibble with columns: activity_id, activity, category, annual_msv, annual_micromorts, milestone columns (mm_Ny for each N), regulatory_limit_msv, xray_equivalents_per_year.

References

ICRP Publication 103 (2007). Recommendations of the International Commission on Radiological Protection.

Brenner DJ, Hall EJ (2007). ”Computed Tomography — An Increasing Source of Radiation Exposure.” NEJM 357:2277-2284.

UNSCEAR 2020. Sources, Effects and Risks of Ionizing Radiation.

See Also

`atomic_risks()`, `patient_radiation_comparison()`

Examples

```
radiation_profiles()
radiation_profiles(milestones = c(5, 25, 50))
```

`ranking_quiz_questions`

Generate Ranking Quiz Questions

Description

Creates questions for the ranking quiz by combining acute (micromort) and chronic (microlife) risks, converting to a common Loss of Life Expectancy (LLE) scale, and grouping into rankable sets.

Usage

```
ranking_quiz_questions(
  tags = NULL,
  items_per_question = 3L,
  n_questions = 5L,
  seed = NULL,
```

```
  difficulty = NULL,
  profile = list()
)
```

Arguments

`tags`	Character vector of tags to include (e.g. "Radiation", "Travel"). Use `NULL` for all tags. See `ranking_tag_mapping()` for available tags.
`items_per_question`	Integer. Number of items per question (2, 3, or 4). Default 3.
`n_questions`	Integer. Number of questions to generate. Default 5.
`seed`	Optional integer seed for reproducibility.
`difficulty`	Optional difficulty level: "easy", "medium", "hard", or "mixed". Easy = large LLE spread within question, hard = small spread.
`profile`	A named list of condition variables for filtering conditional risks, passed to `common_risks()`. E.g. `list(country = "NG")` to include Nigerian disease mortality in the acute risk pool. Default `list()`.

Value

A tibble with columns:

- `question_id`, `tag`, `item_name`, `item_source` ("acute"/"chronic"), `lle_minutes`, `micromorts`, `microlives_per_day`, `category`, `description`, `help_url`, `correct_rank`, `difficulty`

Examples

```
ranking_quiz_questions(tags = "Travel", n_questions = 3, seed = 42)
```

`ranking_tag_mapping` *Tag-to-Category Mapping for Ranking Quiz*

Description

Returns the mapping between user-facing quiz tags and dataset categories. Tags group related risks across both acute (micromort) and chronic (microlife) datasets for the ranking quiz.

Usage

```
ranking_tag_mapping()
```

Value

A tibble with columns `tag`, `source` ("acute"/"chronic"), `category`, and optionally `pattern` (regex for activity-level filtering).

Examples

```
ranking_tag_mapping()
```

`regional_life_expectancy` *Regional Life Expectancy in Western Europe*

Description

Life expectancy at birth by NUTS2 region for Western European countries, based on Eurostat data and the methodology from Bonnet et al. (2026).

Usage

```
regional_life_expectancy(
  country = NULL,
  year = NULL,
  sex = NULL,
  classification = NULL
)
```

Arguments

`country`	Character vector. Filter to specific countries using ISO 2-letter codes (e.g., "FR", "DE", "ES"). Default `NULL` returns all countries.
`year`	Integer or vector. Filter to specific years. Default `NULL` returns all years (1992-2023).
`sex`	Character. Filter by sex: "Male", "Female", or "Total". Default `NULL` returns all.
`classification`	Character. Filter by region classification: "vanguard", "average", or "laggard". Default `NULL` returns all.

Details

Data Structure: Aggregated Population Statistics:

Each row represents one region-year-sex combination, NOT individual survey responses. For example, a dataset with 450 regions × 28 years × 3 sex categories = 37,800 rows of aggregated statistics.

region_code	year	sex	life_expectancy	Meaning
FR10	2019	Male	82.5	Avg LE for all males in Île-de-France in 2019
FR10	2019	Female	87.1	Avg LE for all females in Île-de-France in 2019
FR10	2019	Total	84.8	Avg LE for entire population of Île-de-France in 2019

The underlying Eurostat data represents ~**400 million people** across Western Europe. Life expectancy is calculated from official death registrations and census population counts—not a sample survey.

Data Source:

Primary data from Eurostat dataset `demo_r_mlifexp`. Regional classifications based on Bonnet et al. (2026) methodology identifying:

- **Vanguard regions**: Top 20% life expectancy with sustained gains (1.5 months/year)
- **Laggard regions**: Bottom 20% life expectancy or stagnant gains (<0.5 months/year)
- **Average regions**: All others

Microlives Interpretation:

The `microlives_vs_eu_avg` column converts life expectancy differences to daily microlives using the approximation: 1 year LE difference 1.2 microlives/day (assuming 40 years remaining life expectancy).

Example: A region with +2 years above EU average = +2.4 microlives/day, equivalent to the benefit of 20 minutes daily exercise.

Ecological Fallacy Warning:

IMPORTANT: Regional life expectancy reflects population averages, NOT individual-level causation. High life expectancy in ”vanguard” regions results from multiple factors including:

- Healthcare system quality and access
- Socioeconomic composition (income, education)
- Selection effects (healthy/wealthy people moving to certain regions
- Historical and cultural factors

Moving to a high-LE region does NOT guarantee increased personal longevity.

Value

A tibble with columns:

region_code NUTS2 region code (e.g., ”FR10” for Île-de-France)

region_name Human-readable region name

country_code ISO 2-letter country code

year Data year (1992-2023)

sex Sex category: ”Male”, ”Female”, or ”Total”

life_expectancy Life expectancy at birth in years

microlives_vs_eu_avg Daily microlives difference vs EU average

classification ”vanguard”, ”average”, or ”laggard” based on 2019 trends

source_url DOI link to source publication

References

Bonnet F, et al. (2026). ”Potential and challenges for sustainable progress in human longevity.” Nature Communications 17, 996. doi:10.1038/s4146702668828z

Eurostat (2024). Life expectancy by age, sex and NUTS 2 region (demo_r_mlifexp). `https://ec.europa.eu/eurostat/databrowser/product/view/demo_r_mlifexp`

See Also

`demographic_factors()`, `chronic_risks()`

Other regional: `laggard_regions()`, `regional_mortality_multiplier()`, `vanguard_regions()`

Examples

```
# All data
regional_life_expectancy()

# French regions in 2019
regional_life_expectancy(country = "FR", year = 2019)

# Compare vanguard vs laggard regions
regional_life_expectancy(year = 2019, sex = "Total") |>
  dplyr::group_by(classification) |>
  dplyr::summarise(mean_le = mean(life_expectancy))

# Top 10 regions by life expectancy (2019, Total)
regional_life_expectancy(year = 2019, sex = "Total") |>
  dplyr::slice_max(life_expectancy, n = 10)

# Microlives advantage of Catalonia vs EU average
regional_life_expectancy(country = "ES", year = 2019, sex = "Total") |>
  dplyr::filter(grepl("Catalonia", region_name))
```

`regional_mortality_multiplier` *Regional Mortality Multiplier*

Description

Calculate a mortality risk multiplier for a region relative to the national or EU average. Useful for adjusting baseline micromort estimates by location.

Usage

```
regional_mortality_multiplier(region_code, reference = "eu", year = 2019)
```

Arguments

`region_code`	Character. NUTS2 region code (e.g., ”FR10”).
`reference`	Character. Compare against ”national” average or ”eu” average. Default is ”eu”.
`year`	Integer. Reference year. Default is 2019 (pre-COVID).

Details

The mortality multiplier is derived from life expectancy differences using the approximation that each year of life expectancy difference corresponds to approximately 2.5% difference in annual mortality risk.

A multiplier of 1.0 means average risk; 0.9 means 10% lower risk; 1.1 means 10% higher risk.

Value

A tibble with the region’s mortality multiplier and interpretation.

See Also

`regional_life_expectancy()`, `demographic_factors()`

Other regional: `laggard_regions()`, `regional_life_expectancy()`, `vanguard_regions()`

Examples

```
# Catalonia vs EU average
regional_mortality_multiplier("ES51")

# Compare to national average
regional_mortality_multiplier("ES51", reference = "national")
```

`risk_components` *View Risk Components for an Activity*

Description

Returns the atomic risk components for a specified activity, optionally filtered by health profile. Useful for understanding what contributes to a composite risk value.

Usage

```
risk_components(activity_id, profile = list(), risks = NULL)
```

Arguments

`activity_id`	Character. The activity ID (e.g., `"flying_8h"`). Use `atomic_risks()$activity_id` to see available IDs.
`profile`	A named list of condition variables, e.g. `list(health_profile = "dvt_risk_factors")`.
`risks`	Optional pre-computed `atomic_risks()` tibble.

Value

A tibble of atomic components for the requested activity.

See Also

`atomic_risks()`, `common_risks()`

Examples

```
risk_components("flying_8h")
risk_components("flying_8h", profile = list(health_profile = "dvt_risk_factors"))
```

`risk_data_sources` *Data Sources for Micromort and Microlife Research*

Description

Returns a tibble of authoritative data sources for mortality risk research.

Usage

```
risk_data_sources()
```

Value

A tibble with source names, URLs, types, and descriptions.

Examples

```
risk_data_sources()
```

`risk_equivalence` *Risk Equivalence Table*

Description

Compares a reference activity to all other activities by computing the ratio of micromorts. ”How many X-rays equal one skydive?”

Usage

```
risk_equivalence(reference, risks = NULL, min_ratio = 0.01, max_ratio = Inf)
```

Arguments

`reference`	Character. Activity name to use as the reference (denominator). Must match an `activity` value in `risks`.
`risks`	A tibble with at least `activity` and `micromorts` columns. Defaults to `common_risks()`.
`min_ratio`	Numeric. Minimum ratio to include (default 0.01).
`max_ratio`	Numeric. Maximum ratio to include (default `Inf`).

Value

A tibble with columns: `activity`, `micromorts`, `reference`, `reference_micromorts`, `ratio`, `equivalence`.

See Also

`risk_exchange_matrix()`

Examples

```
risk_equivalence("Chest X-ray (radiation per scan)")
risk_equivalence("Skydiving (US)")
```

`risk_exchange_matrix` *Risk Exchange Matrix*

Description

Creates a cross-comparison matrix showing how many of activity B equal one of activity A, for a selected set of activities.

Usage

```
risk_exchange_matrix(activities = NULL, risks = NULL)
```

Arguments

`activities`	Character vector of activity names to include. Defaults to a curated set of 10 diverse activities.
`risks`	A tibble with at least `activity` and `micromorts` columns. Defaults to `common_risks()`.

Value

A tibble where rows are activities and columns are exchange rates. Cell (i, j) = ”how many of activity j equal one of activity i”.

See Also

`risk_equivalence()`

Examples

```
risk_exchange_matrix()
```

`risk_for_duration` *Calculate Risk for Custom Duration*

Description

For duration-dependent activities, finds the nearest pre-computed duration bucket across all variants of an activity family and returns the aggregated risk.

Usage

```
risk_for_duration(
  activity_prefix,
  duration_hours,
  profile = list(),
  risks = NULL
)
```

Arguments

`activity_prefix`	Character. Activity family prefix (e.g., `"flying"` matches `flying_2h`, `flying_5h`, `flying_8h`, `flying_12h`). Also accepts a full `activity_id`.
`duration_hours`	Numeric. Desired duration in hours.
`profile`	A named list of condition variables.
`risks`	Optional pre-computed `atomic_risks()` tibble.

Value

A tibble with one row per component at the nearest duration bucket, plus summary columns.

See Also

`risk_components()`, `common_risks()`

Examples

```
risk_for_duration("flying", duration_hours = 7)
risk_for_duration("flying", duration_hours = 3)
```

`risk_sensitivity` *Sensitivity Analysis for Activity Risk Estimates*

Description

Computes how activity micromort rankings shift when the base estimate is varied by ±`pct`%. Useful for communicating uncertainty around point estimates derived from sparse epidemiological data.

Usage

```
risk_sensitivity(activity = NULL, pct = 20)
```

Arguments

`activity`	Character scalar — activity name matching a row in `common_risks()`. Pass `NULL` (default) to return sensitivity for all activities.
`pct`	Numeric scalar — percentage variation applied symmetrically around the base estimate. Default `20` (i.e., ±20%). Must be in (0, 100).

Details

Activities are sourced from `common_risks()`. The `rank_change` column reports the absolute number of ranking positions an activity moves between its low and high estimate when all activities are re-ranked.

Value

A tibble with columns:

activity Activity name

micromorts_base Base micromort estimate from `common_risks()`

micromorts_low Low estimate: `base * (1 - pct/100)`

micromorts_high High estimate: `base * (1 + pct/100)`

rank_base Rank of the activity at the base estimate (1 = highest risk)

rank_change Absolute rank positions shifted between low and high estimates

See Also

`common_risks()`, `daily_hazard_rate()`

Other analysis: `annual_risk_budget()`, `compare_interventions()`, `daily_hazard_rate()`, `lifestyle_tradeoff()`, `toxicological_risk()`

Examples

```
# Sensitivity for a single activity
risk_sensitivity("Skydiving (US)")

# Sensitivity for all activities at ±10%
risk_sensitivity(pct = 10)

# Activities with the largest rank uncertainty
risk_sensitivity() |> dplyr::arrange(dplyr::desc(rank_change))
```

`risk_sources` *Risk Sources Registry*

Description

A registry of data sources used to compile the risk datasets. Each source has a unique identifier that links to records in acute_risks and chronic_risks.

Format

A tibble with 14 rows and 7 columns:

source_id Unique source identifier (e.g., ”spiegelhalter_2012”)

citation Full citation or source name

primary_url Primary URL

type Source type: academic, government, database, book, encyclopedia

description Brief description

data_types Types of data: acute, chronic, or both

last_accessed Date data was retrieved

See Also

Other datasets: `acute_risks`, `chronic_risks()`

Examples

```
# Load the source registry
sources <- load_sources()
sources

# Academic sources
sources |> dplyr::filter(type == "Academic")
```

`theme_micromort_dark` *Dark Theme for Micromort Risk Plots*

Description

A dark-background ggplot2 theme designed for risk comparison plots. White text on `#1a1a1a` background with subtle grid lines.

Usage

```
theme_micromort_dark(label_size = 9)
```

Arguments

`label_size`	Numeric. Y-axis label font size. Default is 9.

Value

A ggplot2 theme object.

See Also

Other visualization: `plot_risks()`, `plot_risks_interactive()`, `prepare_risks_plot()`

Examples

```
library(ggplot2)
ggplot(mtcars, aes(mpg, wt)) + geom_point(color = "white") + theme_micromort_dark()
```

`toxicological_risk` *Calculate micromorts from substance exposure using LD50 data*

Description

Uses human LD50 estimates from `inst/extdata/ld50_human.csv` to translate a dose into a micromort risk estimate via linear extrapolation from the LD50 reference point. At the LD50, 50% lethality equals 500,000 micromorts.

Usage

```
toxicological_risk(substance = NULL, dose_mg = NULL, body_weight_kg = 70)
```

Arguments

`substance`	Character scalar. Name of substance (case-insensitive, partial match supported). Pass `NULL` (default) to return the full reference table of all substances with their LD50 values.
`dose_mg`	Numeric scalar. Dose in milligrams. Required when `substance` is specified; ignored otherwise.
`body_weight_kg`	Numeric scalar. Body weight in kg. Default `70`.

Value

A tibble. When `substance = NULL`, returns all substances with columns `substance`, `route`, `ld50_mg_per_kg`, `source`. When a substance is specified, returns a single-row tibble with additional columns: `dose_mg`, `fraction_of_ld50`, `micromorts`, `risk_category`.

See Also

`common_risks()`, `risk_sensitivity()`

Other analysis: `annual_risk_budget()`, `compare_interventions()`, `daily_hazard_rate()`, `lifestyle_tradeoff()`, `risk_sensitivity()`

Examples

```
# Full reference table
toxicological_risk()

# Risk from 1 mg nicotine for a 70 kg person
toxicological_risk("Nicotine", dose_mg = 1)

# Partial name matching
toxicological_risk("nico", dose_mg = 1)

# Different body weight
toxicological_risk("Caffeine", dose_mg = 200, body_weight_kg = 80)
```

`vaccination_risks` *Vaccination Risk Reduction*

Description

Mortality risk reduction from vaccination schedules compared to unvaccinated baseline, expressed in micromorts avoided per year.

Usage

```
vaccination_risks()
```

Details

Data from CDC, WHO, and Lancet 2024 Global Vaccine Impact Study.

Value

A tibble with vaccination schedules and their risk reduction metrics.

References

CDC. Health and Economic Benefits of Routine Childhood Immunizations. MMWR 2024;73:1-8. https://www.cdc.gov/mmwr/

Lancet 2024. Contribution of vaccination to improved survival: 50 years of EPI. doi:10.1016/S01406736(24)00850X

See Also

`cancer_risks()`, `conditional_risk()`, `hedged_portfolio()`

Other conditional-risk: `cancer_risks()`, `conditional_risk()`, `hedged_portfolio()`

Examples

```
vaccination_risks()
vaccination_risks() |> dplyr::filter(country == "US")
vaccination_risks() |> dplyr::filter(age_group == "0-5")  # Childhood vaccines
```

`value_of_micromort` *Value of a Statistical Life (VSL) to Micromort Value*

Description

Calculates the monetary value of one micromort based on the Value of a Statistical Life (VSL).

Usage

```
value_of_micromort(vsl = 1e+07)
```

Arguments

`vsl`	Numeric. Value of a Statistical Life (default $10,000,000).

Value

Numeric value of one micromort.

See Also

`as_micromort()`, `lle()`

Other conversion: `as_microlife()`, `as_micromort()`, `as_probability()`, `lle()`

Examples

```
value_of_micromort(10000000) # $10
```

`vanguard_regions` *Vanguard Regions with Highest Life Expectancy*

Description

Convenience function returning regions classified as ”vanguard” - those with the highest life expectancy and sustained improvement trends.

Usage

```
vanguard_regions(country = NULL, year = NULL, sex = NULL)
```

Arguments

`country`	Character vector. Filter to specific countries using ISO 2-letter codes (e.g., ”FR”, ”DE”, ”ES”). Default `NULL` returns all countries.
`year`	Integer or vector. Filter to specific years. Default `NULL` returns all years (1992-2023).
`sex`	Character. Filter by sex: ”Male”, ”Female”, or ”Total”. Default `NULL` returns all.

Value

A tibble filtered to vanguard regions only.

See Also

`regional_life_expectancy()`, `laggard_regions()`

Other regional: `laggard_regions()`, `regional_life_expectancy()`, `regional_mortality_multiplier()`

Examples

```
# Vanguard regions in 2019
vanguard_regions(year = 2019, sex = "Total")
```

Index