Package 'expoquimR'

November 14, 2026

Title Qualitative and Quantitative Assessment of Occupational Chemical Exposure Risk

Version 0.2.1

Description Provides a unified toolkit for occupational chemical exposure risk assessment, implementing three internationally recognised methods end to end: the qualitative control-banding methods COSHH Essentials (UK Health and Safety Executive) and the method of the French National Research and Safety Institute (INRS), together with the quantitative statistical procedure of the UNE-EN 689 standard for comparing measured exposure levels against occupational exposure limits. Every step of each method, from hazard banding and exposure scoring to lognormal or normal distribution fitting, one-sided tolerance limits, and monitoring-interval recommendations, is implemented as a small, independently callable, and unit-tested function, so assessments are reproducible and auditable without depending on any graphical interface. Optional 'shiny' applications provide a guided, interactive workflow for occupational hygienists and health and safety practitioners who prefer not to write code.
References: UK Health and Safety Executive (2003) <https://www.hse.gov.uk/coshh/essentials/index.htm>;
Mallet, Pilorget and Berne (2013, ISBN:978-2-7389-2166-2) ``Méthodologie d'évaluation simplifiée du risque chimique: un outil d'aide à la décision'' INRS ND 2233;
European Committee for Standardisation (2018) <https://www.en-standard.eu/bs-en-689-2018-workplace-exposure-measurement-of-exposure-by-inhalation-to-chemical-agents-strate

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

URL https://github.com/Aguilar-Elena/expoquimR, https://aguilar-elena.github.io/expoquimR/

BugReports https://github.com/Aguilar-Elena/expoquimR/issues

Depends R (>= 4.1.0)

Imports stats

Suggests DT, ggplot2, kableExtra, knitr, readxl, rmarkdown, shiny, shinyjs, testthat (>= 3.0.0), tibble, usethis

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Raúl Aguilar-Elena [aut, cre] (ORCID: <https://orcid.org/0000-0001-8656-8155>), Ana Delgado-Garcia [aut] (ORCID: <https://orcid.org/0009-0003-7079-5507>), Alejandro Guillem-Riquelme [aut] (ORCID: <https://orcid.org/0000-0002-6373-5237>)

Maintainer Raúl Aguilar-Elena <raguilar@universidadviu.com>

Repository CRAN

Date/Publication 2026-11-14 15:40:18 UTC

Contents

coshh_classify_volatility . . . 3
coshh_evaluate . . . 4
coshh_from_excel . . . 5
coshh_grade . . . 6
coshh_measures . . . 7
coshh_risk . . . 7
expoquimr_lang . . . 8
expoquimr_report . . . 9
inrs_collective_protection . . . 10
inrs_evaluate . . . 11
inrs_frequency_class . . . 13
inrs_from_excel . . . 13
inrs_hazard_class . . . 14
inrs_inhalation_risk . . . 15
inrs_liquid_volatility_graph . . . 16
inrs_liquid_volatility_pressure . . . 16
inrs_oel_correction_factor . . . 17
inrs_potential_exposure_class . . . 17
inrs_potential_risk_class . . . 18
inrs_potential_risk_score . . . 19
inrs_process_type . . . 19
inrs_quantity_class . . . 20
inrs_risk_characterisation . . . 20
inrs_solid_dustiness . . . 21
inrs_volatility_score . . . 21

run_coshh . . . 22
run_inrs . . . 23
run_une689 . . . 23
une689_classify_conformity . . . 24
une689_daily_exposure . . . 25
une689_distribution_type . . . 25
une689_evaluate_preliminary . . . 26
une689_evaluate_statistical . . . 27
une689_exposure_index . . . 28
une689_from_excel . . . 28
une689_lsc . . . 29
une689_monitoring_interval_opt1 . . . 30
une689_monitoring_interval_opt2 . . . 31
une689_normality_test . . . 31
une689_probability_data . . . 32
une689_probability_plot . . . 34
une689_statistical_conformity . . . 35
une689_statistics . . . 36
une689_ur . . . 36
une689_ut . . . 37
une689_validate_min_days . . . 38

Index **39**

`coshh_classify_volatility`

Classify the volatility of a liquid using the COSHH Essentials method

---

Description

Compares the boiling point of a substance with its process temperature to assign a volatility class (`"Low"` / `"Medium"` / `"High"` in English, `"Baja"` / `"Media"` / `"Alta"` in Spanish), following the thresholds of the COSHH Essentials method.

Usage

```
coshh_classify_volatility(boiling_point, process_temp)
```

Arguments

`boiling_point`	Numeric. Boiling point of the substance, in degrees Celsius.
`process_temp`	Numeric. Temperature at which the substance is handled, in degrees Celsius.

Details

The active language is controlled by `expoquimr_lang()`.

Value

Character scalar: volatility class in the active language.

Examples

```
coshh_classify_volatility(boiling_point = 111, process_temp = 20)
expoquimr_lang("es")
coshh_classify_volatility(boiling_point = 111, process_temp = 20)
expoquimr_lang("en")
```

`coshh_evaluate` *Evaluate a substance using the COSHH Essentials method (high-level wrapper)*

Description

Chains `coshh_grade()`, `coshh_classify_volatility()` (if applicable), `coshh_risk()` and `coshh_measures()` to produce a complete result row from the raw data of a substance. Designed to be called directly from code (scripts, reports, `purrr::pmap`, vignettes) without going through the Shiny application.

Usage

```
coshh_evaluate(
  name,
  phrases,
  quantity,
  is_liquid,
  boiling_point = NA_real_,
  process_temp = NA_real_,
  dustiness = NA_character_
)
```

Arguments

`name`	Character. Identifying name of the substance.
`phrases`	Character. H/R phrases; see `coshh_grade()`.
`quantity`	Character. Quantity class in the active language.
`is_liquid`	Logical. `TRUE` if the substance is liquid (volatility will be calculated from `boiling_point` / `process_temp`); `FALSE` if solid (`dustiness` will be used directly).
`boiling_point, process_temp`	Numeric. Required only if `is_liquid = TRUE`. See `coshh_classify_volatility()`.
`dustiness`	Character. Required only if `is_liquid = FALSE`. Dustiness class in the active language.

Details

Output labels (volatility class, quantity class, control measures) are returned in the active language; see `expoquimr_lang()`.

Value

A one-row `data.frame` with columns `substance`, `phrases`, `grade`, `volatility`, `quantity`, `risk` and `measures`.

Examples

```
coshh_evaluate(
  name = "Toluene",
  phrases = "H315, H336",
  quantity = "Medium",
  is_liquid = TRUE,
  boiling_point = 111,
  process_temp = 20
)
```

---

coshh_from_excel *Evaluate COSHH substances from an Excel file*

Description

Reads an Excel sheet with the format of the expoquimR COSHH template (one row per substance) and returns the complete assessment of each one by calling `coshh_evaluate()`.

Usage

```
coshh_from_excel(path, sheet = "COSHH_datos")
```

Arguments

`path`	Character. Path to the `.xlsx` file. Can be obtained with `system.file()` for the template included in the package, or be any local path.
`sheet`	Character or integer. Name or number of the sheet that contains the data (default `"COSHH_datos"`).

Value

A `data.frame` with one row per substance and the result columns of `coshh_evaluate()`.

Examples

```
# With the template included in the package:
path <- system.file("plantillas", "plantilla_coshh.xlsx",
                    package = "expoquimR")
coshh_from_excel(path)

# With your own file, simply pass its path:
# coshh_from_excel("my_coshh_data.xlsx")
```

`coshh_grade` *Determine the COSHH hazard group from R/H phrases*

Description

Looks up each risk phrase (R) or hazard phrase (H) in the COSHH Essentials hazard group assignment table (groups A to E) and returns the most unfavourable group found. Any phrase not listed explicitly in groups B-E is assigned to group A, following the default rule of the original method.

Usage

```
coshh_grade(phrases)
```

Arguments

`phrases`	Character scalar containing one or more phrases separated by commas, e.g. `"H315, H319"` or `"R20/21/22"`.

Value

Character scalar with the group (`"A"` to `"E"`), or `NA_character_` if `phrases` is empty or `NA`.

Examples

```
coshh_grade("H315, H319")
coshh_grade("R23/24/25")
```

coshh_measures *Get the recommended control measures for a COSHH risk level*

Description

Get the recommended control measures for a COSHH risk level

Usage

```
coshh_measures(risk_level)
```

Arguments

`risk_level`	Integer or character. Risk level (1 to 4), as returned by `coshh_risk()`.

Value

Character scalar with the recommended control measures in the active language (see `expoquimr_lang()`), or `NA_character_` if the level is not defined.

Examples

```
coshh_measures(3)
expoquimr_lang("es")
coshh_measures(3)
expoquimr_lang("en")
```

coshh_risk *Calculate the COSHH risk level*

Description

Queries the hazard x quantity x volatility matrix of the COSHH Essentials method to obtain the potential risk level (1 to 4). Substances of grade `"E"` (carcinogenic, mutagenic or similar) always receive the maximum level, regardless of quantity or volatility.

Usage

```
coshh_risk(grade, quantity, volatility)
```

Arguments

grade	Character. Hazard group ("A" to "E"), as returned by coshh_grade().
quantity	Character. Quantity class in the active language. Use "Small" / "Medium" / "Large" (English) or "Pequeña" / "Mediana" / "Grande" (Spanish).
volatility	Character. Volatility class in the active language. Use "Low" / "Medium" / "High" (English) or "Baja" / "Media" / "Alta" (Spanish). For solids, this corresponds to dustiness.

Value

Integer with the risk level (1 to 4), or NA_integer_ if the combination is not defined in the table or grade is NA.

Examples

```
coshh_risk(grade = "C", quantity = "Medium", volatility = "High")
coshh_risk(grade = "E", quantity = "Small",  volatility = "Low")
```

---

expoquimr_lang *Get or set the language used by expoquimR*

Description

expoquimR supports English ("en", default) and Spanish ("es"). The active language controls the language of function output messages, column labels returned by high-level wrapper functions, and error/warning messages. It does **not** affect the Shiny apps, which have their own in-app language selector.

Usage

```
expoquimr_lang(lang = NULL)
```

Arguments

lang	Character. "en" (English, default) or "es" (Spanish). If NULL, returns the currently active language without changing it.

Details

The active language is stored in a private package environment, not in options(), so calling this function never alters the user's global R session settings.

Value

Invisibly returns the previously active language.

Examples

```
expoquimr_lang()          # query current language
old <- expoquimr_lang("es")  # switch to Spanish, saving the previous value
expoquimr_lang(old)          # restore it
```

`expoquimr_report` *Generate an occupational chemical exposure risk assessment report*

Description

Renders a self-contained HTML report from the results of one or more assessment methods implemented in expoquimR. Each method section is included only when the corresponding argument is supplied. The report includes input data, result tables, density plots (UNE-EN 689), additive effect groups, and a citation block for the package.

Usage

```
expoquimr_report(
  coshh = NULL,
  inrs = NULL,
  une689 = NULL,
  evaluator = "",
  workplace = "",
  output = "expoquimr_report.html",
  lang = .expoquimR_state$lang,
  open = TRUE
)
```

Arguments

`coshh`	A `data.frame` returned by `coshh_evaluate()` or `coshh_from_excel()`. If `NULL` (default), the COSHH section is omitted.
`inrs`	A `data.frame` returned by `inrs_evaluate()` or `inrs_from_excel()`. If `NULL` (default), the INRS section is omitted.
`une689`	A list returned by `une689_from_excel()` or constructed manually with elements `$preliminary` and optionally `$additive`. If `NULL` (default), the UNE-EN 689 section is omitted.
`evaluator`	Character. Name of the person responsible for the assessment. Displayed in the report header. Default `""`.
`workplace`	Character. Name or description of the workplace or workstation assessed. Default `""`.
`output`	Character. Path and filename for the output HTML file. Default `"expoquimr_report.html"` in the current working directory.

lang	Character. Language for the report body: "en" (English, default) or "es" (Spanish). See expoquimr_lang().
open	Logical. Whether to open the report in the default browser after rendering. Default TRUE.

Value

Invisibly returns the path to the generated HTML file.

Examples

```
# COSHH only
res <- coshh_evaluate(
  name = "Toluene", phrases = "H315, H336",
  quantity = "Medium", is_liquid = TRUE,
  boiling_point = 111, process_temp = 20
)
out1 <- tempfile(fileext = ".html")
expoquimr_report(
  coshh     = res,
  evaluator = "Dr. Jane Smith",
  workplace = "Printing workshop A",
  output    = out1,
  open      = FALSE
)

# All three methods
path <- system.file("plantillas", "plantilla_une689.xlsx",
                    package = "expoquimR")
res_une <- une689_from_excel(path)

out2 <- tempfile(fileext = ".html")
expoquimr_report(
  coshh     = res,
  une689    = res_une,
  evaluator = "Dr. Jane Smith",
  workplace = "Printing workshop A",
  output    = out2,
  lang      = "en",
  open      = FALSE
)
```

---

inrs_collective_protection

Collective protection class and score (INRS method)

Description

Collective protection class and score (INRS method)

Usage

```
inrs_collective_protection(situation)
```

Arguments

situation	Character. One of the situations of INRS Figure 4, e.g. "Enclosing hood / full enclosure".

Value

A one-row data.frame with columns class and score.

Examples

```
inrs_collective_protection("Enclosing hood / full enclosure")
```

---

inrs_evaluate *Evaluate a chemical product with the INRS method (high-level wrapper)*

Description

Chains all the steps of the INRS method (hazard class, quantity, frequency, potential exposure, potential risk, volatility or dustiness, process and collective protection) from the raw data of a product, and returns a complete result row. Designed to be used directly from code, without going through the Shiny application.

Usage

```
inrs_evaluate(
  name,
  r_phrases = character(0),
  h_phrases = character(0),
  process = NULL,
  vla = NA_real_,
  quantity_value = NA_real_,
  quantity_unit = c("g", "ml", "kg", "l"),
  frequency_value = NA_real_,
  frequency_unit = c("minutes", "hours", "days", "months", "not_used"),
  substance_type = c("liquid", "solid"),
  liquid_method = c("graph", "pressure"),
  use_temperature = NA_real_,
```

```
  boiling_point = NA_real_,
  vapour_pressure = NA_real_,
  solid_description = NA_character_,
  procedure,
  protection
)
```

Arguments

`name`	Character. Name of the product.
`r_phrases`, `h_phrases`	Character vectors. See `inrs_hazard_class()`.
`process`	Character. See `inrs_hazard_class()`.
`vla`	Numeric. VLA in mg/m3.
`quantity_value`, `quantity_unit`	See `inrs_quantity_class()`.
`frequency_value`, `frequency_unit`	See `inrs_frequency_class()`.
`substance_type`	Character. `"liquid"` or `"solid"`.
`liquid_method`	Character. `"graph"` or `"pressure"`. Only used if `substance_type = "liquid"`.
`use_temperature`, `boiling_point`	Numeric. Only if `liquid_method = "graph"`.
`vapour_pressure`	Numeric. Only if `liquid_method = "pressure"`.
`solid_description`	Character. Only if `substance_type = "solid"`. See `inrs_solid_dustiness()`.
`procedure`	Character. See `inrs_process_type()`.
`protection`	Character. See `inrs_collective_protection()`.

Value

A one-row `data.frame` with all the intermediate classes and scores, the final inhalation risk score and its characterisation.

Examples

```
inrs_evaluate(
  name = "Solvent X",
  h_phrases = "H336",
  vla = 50,
  quantity_value = 5, quantity_unit = "l",
  frequency_value = 3, frequency_unit = "hours",
  substance_type = "liquid",
  liquid_method = "graph",
  use_temperature = 40, boiling_point = 80,
  procedure = "Open",
  protection = "Moderate dispersion conditions"
)
```

inrs_frequency_class *Frequency of use class (INRS method)*

Description

Converts the given frequency of use to the reference units of Table 3 of the INRS method (hours/day or days/month or days/year, depending on the input unit) and returns the corresponding class (0 to 4).

Usage

```
inrs_frequency_class(
  value = NA_real_,
  unit = c("minutes", "hours", "days", "months", "not_used")
)
```

Arguments

value	Numeric. Frequency value. Ignored if `unit = "not_used"`.
unit	Character. One of `"minutes"`, `"hours"`, `"days"`, `"months"`, `"not_used"` (the latter for substances that are not used with a periodic frequency, and always returns class `"0"`).

Value

Character scalar (`"0"` to `"4"`), or `NA_character_` if there is no match in the reference table.

Examples

```
inrs_frequency_class(3, "hours")
inrs_frequency_class(unit = "not_used")
```

inrs_from_excel *Evaluate INRS chemical products from an Excel file*

Description

Reads an Excel sheet with the format of the expoquimR INRS template (one row per product) and returns the complete assessment by calling `inrs_evaluate()`.

Usage

```
inrs_from_excel(path, sheet = "INRS_datos")
```

Arguments

path	Character. Path to the .xlsx file.
sheet	Character or integer. Name or number of the sheet (default "INRS_datos").

Value

A data.frame with one row per product and all the result columns of inrs_evaluate().

Examples

```
path <- system.file("plantillas", "plantilla_inrs.xlsx",
                    package = "expoquimR")
inrs_from_excel(path)
```

---

inrs_hazard_class *Hazard class of a substance (INRS method)*

Description

Determines the hazard class (1 to 5) of a substance from its R phrases, its H phrases, its VLA, or the material/process it belongs to, by consulting Table 1 of the INRS method. It is explored from the most hazardous class (5) down to the least hazardous (1), and the first class with a match is returned. If no phrase, VLA or process matches classes 2 to 5, class 1 is assigned by default (catch-all: "has phrases but none of the above"), as established by the INRS methodology.

Usage

```
inrs_hazard_class(
  r_phrases = character(0),
  h_phrases = character(0),
  process = NULL,
  vla = NA_real_
)
```

Arguments

r_phrases	Character vector of R phrases (optional).
h_phrases	Character vector of H phrases (optional).
process	Character scalar with the material/process (optional). Compared as a substring (case-insensitive) against the text of Table 1.
vla	Numeric. VLA in mg/m3 (optional).

Value

Character scalar ("1" to "5"). Only returns NA_character_ if no criterion at all was supplied (r_phrases, h_phrases, process and vla all empty/NA), in which case there is not enough information to classify.

Examples

```
inrs_hazard_class(h_phrases = "H335")
inrs_hazard_class(vla = 0.05)
inrs_hazard_class(vla = 200) # falls into class 1 by default
```

---

inrs_inhalation_risk *Final inhalation risk score (INRS method)*

Description

Product of the five partial scores of the INRS method.

Usage

```
inrs_inhalation_risk(
  potential_risk_score,
  volatility_score,
  procedure_score,
  protection_score,
  vla_correction_factor
)
```

Arguments

potential_risk_score	Numeric. See inrs_potential_risk_score().
volatility_score	Numeric. See inrs_volatility_score().
procedure_score	Numeric. See inrs_process_type().
protection_score	Numeric. See inrs_collective_protection().
vla_correction_factor	Numeric. See inrs_oel_correction_factor().

Value

Numeric, or NA_real_ if any component is missing.

Examples

```
inrs_inhalation_risk(100, 10, 0.5, 0.7, 10)
```

inrs_liquid_volatility_graph

Volatility class of a liquid from use temperature and boiling point

Description

Classifies the volatility of a liquid by comparing its boiling point with the two class-separating lines of the INRS method graph (Figure 2, use temperature on the X axis, boiling point on the Y axis).

Usage

```
inrs_liquid_volatility_graph(use_temperature, boiling_point)
```

Arguments

use_temperature	Numeric. Use (process) temperature, in degrees Celsius. Corresponds to the X axis of the graph.
boiling_point	Numeric. Boiling point, in degrees Celsius. Corresponds to the Y axis of the graph.

Value

Character scalar (`"1"` low, `"2"` medium, `"3"` high).

Examples

```
inrs_liquid_volatility_graph(use_temperature = 20, boiling_point = 200)
inrs_liquid_volatility_graph(use_temperature = 20, boiling_point = 80)
```

inrs_liquid_volatility_pressure

Volatility class of a liquid from vapour pressure

Description

Classifies the volatility of a liquid according to the official thresholds of Table 8 of the INRS method.

Usage

```
inrs_liquid_volatility_pressure(vapour_pressure)
```

Arguments

vapour_pressure	Numeric. Vapour pressure at working temperature, in kPa.

Value

Character scalar ("1" if Pv < 0.5 kPa, "2" if 0.5 <= Pv < 25 kPa, "3" if Pv >= 25 kPa).

Examples

```
inrs_liquid_volatility_pressure(15)
```

---

inrs_oel_correction_factor

VLA correction factor (INRS method)

Description

VLA correction factor (INRS method)

Usage

```
inrs_oel_correction_factor(vla)
```

Arguments

vla	Numeric. VLA in mg/m3.

Value

Numeric (1, 10, 30 or 100), or NA_real_ if vla is NA.

Examples

```
inrs_oel_correction_factor(0.05)
```

---

inrs_potential_exposure_class

Potential exposure class (INRS method)

Description

Consults Table 4 of the INRS method (quantity class x frequency class) to obtain the potential exposure class.

Usage

```
inrs_potential_exposure_class(quantity_class, frequency_class)
```

Arguments

quantity_class	Character. See inrs_quantity_class().
frequency_class	Character. See inrs_frequency_class().

Value

Character scalar ("0" to "5"), or NA_character_ if the combination is not defined.

Examples

```
inrs_potential_exposure_class("3", "2")
```

---

inrs_potential_risk_class

Potential risk class (INRS method)

Description

Consults Table 5 of the INRS method (potential exposure class x hazard class) to obtain the potential risk class.

Usage

```
inrs_potential_risk_class(potential_exposure_class, hazard_class)
```

Arguments

potential_exposure_class	Character. See inrs_potential_exposure_class().
hazard_class	Character. See inrs_hazard_class().

Value

Character scalar ("1" to "5"), or NA_character_ if the combination is not defined.

Examples

```
inrs_potential_risk_class("4", "3")
```

inrs_potential_risk_score *Potential risk score (INRS method)*

Description

Potential risk score (INRS method)

Usage

```
inrs_potential_risk_score(potential_risk_class)
```

Arguments

potential_risk_class	Character. See inrs_potential_risk_class().

Value

Numeric (1, 10, 100, 1000 or 10000), or NA_real_.

Examples

```
inrs_potential_risk_score("3")
```

inrs_process_type *Process class and score (INRS method)*

Description

Process class and score (INRS method)

Usage

```
inrs_process_type(type)
```

Arguments

type	Character. One of "Dispersive", "Open", "Closed/opened regularly", "Permanently closed".

Value

A one-row data.frame with columns class and score.

Examples

```
inrs_process_type("Open")
```

`inrs_quantity_class` *Daily handled quantity class (INRS method)*

Description

Classifies the daily quantity of substance handled into one of the 5 classes of the INRS method, according to its unit.

Usage

```
inrs_quantity_class(value, unit = c("g", "ml", "kg", "l"))
```

Arguments

`value`	Numeric. Daily quantity handled.
`unit`	Character. One of `"g"`, `"ml"`, `"kg"`, `"l"`.

Value

Character scalar (`"1"` to `"5"`), or `NA_character_` if `value` is `NA`.

Examples

```
inrs_quantity_class(50, "g")
inrs_quantity_class(500, "kg")
```

`inrs_risk_characterisation`
Inhalation risk characterisation (INRS method)

Description

Inhalation risk characterisation (INRS method)

Usage

```
inrs_risk_characterisation(inhalation_risk)
```

Arguments

`inhalation_risk`
Numeric. See `inrs_inhalation_risk()`.

Value

Character scalar describing the action priority, or NA_character_ if inhalation_risk is NA.

Examples

```
inrs_risk_characterisation(2500)
```

---

inrs_solid_dustiness *Dustiness class of a solid (INRS method)*

Description

Dustiness class of a solid (INRS method)

Usage

```
inrs_solid_dustiness(description)
```

Arguments

description	Character. One of "Dust that generates a lot of visible dispersion in the air", "Fine dust with little visible dispersion" or "Compact solid with no visible dust".

Value

Character scalar ("1" to "3"), or NA_character_ if description does not match any valid option.

Examples

```
inrs_solid_dustiness("Fine dust with little visible dispersion")
```

---

inrs_volatility_score *Volatility or dustiness score (INRS method)*

Description

Volatility or dustiness score (INRS method)

Usage

```
inrs_volatility_score(volatility_class)
```

Arguments

volatility_class
: Character. "1", "2" or "3", as returned by inrs_liquid_volatility_graph(), inrs_liquid_volatility_pressure() or inrs_solid_dustiness().

Value

Numeric (1, 10 or 100), or NA_real_.

Examples

```
inrs_volatility_score("2")
```

---

run_coshh *Launch the COSHH Essentials Shiny application*

Description

Launch the COSHH Essentials Shiny application

Usage

```
run_coshh(...)
```

Arguments

...
: Additional arguments passed to shiny::runApp() (e.g. launch.browser, port).

Value

No return value, called for side effects. Launches a Shiny application in the default browser.

Examples

```
run_coshh()
```

run_inrs *Launch the INRS method Shiny application*

Description

Launch the INRS method Shiny application

Usage

```
run_inrs(...)
```

Arguments

...	Additional arguments passed to `shiny::runApp()` (e.g. `launch.browser`, `port`).

Value

No return value, called for side effects. Launches a Shiny application in the default browser.

Examples

```
run_inrs()
```

run_une689 *Launch the UNE-EN 689 Shiny application (preliminary, statistical and periodic assessment)*

Description

Launch the UNE-EN 689 Shiny application (preliminary, statistical and periodic assessment)

Usage

```
run_une689(...)
```

Arguments

...	Additional arguments passed to `shiny::runApp()` (e.g. `launch.browser`, `port`).

Value

No return value, called for side effects. Launches a Shiny application in the default browser.

Examples

```
run_une689()
```

`une689_classify_conformity` *Classify the conformity of the UNE-EN 689 preliminary assessment*

Description

From the exposure indices (IE) of all the days evaluated, determines whether exposure is conforming, non-conforming, or whether no decision can be made without further measurements, following the criteria of the UNE-EN 689 preliminary assessment.

Usage

```
une689_classify_conformity(ie)
```

Arguments

`ie`	Numeric vector. Exposure indices, one per day (see `une689_exposure_index()`). `NA` values (days without enough data) are ignored.

Value

Character scalar: `.t("une689_conformity")` if all IE values are below 0.1; `.t("une689_no_conformity")` if any IE is above 1; `.t("une689_no_decision")` if any IE is between 0.1 and 1 (both inclusive) and none exceeds 1. Returns `NA_character_` if there is no valid IE at all (not enough data to classify).

Correction relative to the original app

In the original Shiny app, if all days had IE = `NA` (due to missing data), the check `all(IEs < 0.1, na.rm = TRUE)` returned `TRUE` (because `all()` over an empty vector is `TRUE` in R), and the result was incorrectly reported as **.t("une689_conformity")** with no actual data. This function fixes that case by returning `NA_character_` (not enough data) instead of a false conformity. This change is flagged here because, unlike the INRS points, it was applied without prior confirmation: reporting "conformity" with no data is a safety flaw, not a methodological judgement call.

Examples

```
une689_classify_conformity(c(0.02))
une689_classify_conformity(c(1, 0.9, 0.56))
une689_classify_conformity(c(1.2, 0.05))
```

une689_daily_exposure *Daily exposure (ED) for a measurement day, per UNE-EN 689*

Description

Calculates the daily exposure from the concentrations and times of the valid samples of a measurement day. If there is a single valid sample taken over the complete 8-hour working day, the ED is directly that concentration. Otherwise, it is calculated as the time-weighted average over an 8-hour day (`sum(concentration * time) / 8`).

Usage

```
une689_daily_exposure(concentration, time)
```

Arguments

`concentration`	Numeric vector. Measured concentrations (mg/m3), one per sample.
`time`	Numeric vector. Time of each sample (hours), same length as `concentration`.

Value

Numeric scalar with the ED, or `NA_real_` if there is no valid (concentration, time) pair.

Examples

```
une689_daily_exposure(concentration = c(12, 8), time = c(4, 4))
une689_daily_exposure(concentration = 9, time = 8)
```

une689_distribution_type *Determine the distribution type (UNE-EN 689)*

Description

Decides whether the data best fits a lognormal, normal, or neither distribution, from the Shapiro-Wilk p-values. Priority is given to the lognormal fit, following common practice in industrial hygiene (exposure is usually lognormal).

Usage

```
une689_distribution_type(pval_normal, pval_lognormal, alpha = 0.05)
```

Arguments

pval_normal	Numeric. p-value of the normality test on ED.
pval_lognormal	Numeric. p-value of the normality test on log(ED).
alpha	Numeric. Significance level (default 0.05).

Value

Character scalar: one of "Lognormal", "Normal" or "Neither" (in English, the default), or the equivalent Spanish labels when expoquimr_lang("es") is active.

Examples

```
une689_distribution_type(pval_normal = 0.03, pval_lognormal = 0.20)
```

---

une689_evaluate_preliminary

Complete UNE-EN 689 preliminary assessment (high-level wrapper)

Description

Calculates the ED and IE for each day and classifies the overall conformity, from a set of samples organised by measurement day. Designed to be used directly from code, without going through the Shiny application.

Usage

```
une689_evaluate_preliminary(data, vla)
```

Arguments

data	A data.frame in long format with columns day (day identifier, numeric or character), concentration (mg/m3) and time (hours). One row per sample.
vla	Numeric. Valor Limite Ambiental / occupational exposure limit (mg/m3).

Value

A list with two elements:

days_table A data.frame with columns day, ED and IE, one row per day.

result Character scalar with the overall classification, see une689_classify_conformity().

Examples

```
data <- data.frame(
  day = c(1, 1, 2, 3, 3),
  concentration = c(12, 8, 9, 5, 6),
  time = c(4, 4, 8, 3, 5)
)
une689_evaluate_preliminary(data, vla = 10)
```

---

une689_evaluate_statistical

Complete UNE-EN 689 statistical assessment (high-level wrapper)

Description

Chains the distribution fit, the calculation of UT, LSC(95,70), UR and the statistical conformity from a set of daily exposure (ED) values. Designed to be used directly from code, without going through the Shiny application.

Usage

```
une689_evaluate_statistical(ed, vla)
```

Arguments

`ed`	Numeric vector. ED values (one per day), all positive. A minimum of 6 is required (see `une689_validate_min_days()`).
`vla`	Numeric. Valor Limite Ambiental / occupational exposure limit (mg/m3).

Value

A list with elements `n`, `distribution_type`, `MA`, `DS`, `MG`, `DSG`, `W_normal`, `pval_normal`, `W_lognormal`, `pval_lognormal`, `ut`, `lsc`, `ur`, `conformity`.

Examples

```
une689_evaluate_statistical(c(5, 6, 7, 8, 9, 10), vla = 10)
```

une689_exposure_index *Exposure index (IE) for a measurement day, per UNE-EN 689*

Description

Exposure index (IE) for a measurement day, per UNE-EN 689

Usage

```
une689_exposure_index(ed, vla)
```

Arguments

ed	Numeric. Daily exposure, see une689_daily_exposure().
vla	Numeric. Valor Limite Ambiental / occupational exposure limit (mg/m3).

Value

Numeric scalar (ed / vla), or NA_real_ if ed or vla are not valid (vla must be > 0).

Examples

```
une689_exposure_index(ed = 9, vla = 10)
```

une689_from_excel *Evaluate UNE-EN 689 chemical exposure from an Excel file*

Description

Reads the three sheets of the expoquimR UNE-EN 689 template (Agents, Measurements and optionally Additive_effects) and returns a list with the preliminary assessment of each agent, and if applicable, the additive effects calculation by group.

Usage

```
une689_from_excel(path)
```

Arguments

path	Character. Path to the .xlsx file.

Value

A list with the elements:

`preliminary` A list with one element per agent, each with `name`, `vla`, `days_table` and `result`.

`additive` `data.frame` with columns `group`, `agent`, `mean_ie` and `combined_ie`, or `NULL` if there is no additive effects sheet.

Examples

```
path <- system.file("plantillas", "plantilla_une689.xlsx",
                    package = "expoquimR")
res <- une689_from_excel(path)
res$preliminary
res$additive
```

`une689_lsc` *Upper confidence limit LSC(95,70) (UNE-EN 689)*

Description

Upper confidence limit LSC(95,70) (UNE-EN 689)

Usage

```
une689_lsc(
  distribution_type,
  ut,
  MA = NA_real_,
  DS = NA_real_,
  MG = NA_real_,
  DSG = NA_real_
)
```

Arguments

`distribution_type`	Character. Distribution type as returned by `une689_distribution_type()`. Accepts both English (`"Lognormal"`, `"Normal"`, `"Neither"`) and Spanish labels.
`ut`	Numeric. UT factor, see `une689_ut()`.
`MA, DS`	Numeric. Arithmetic mean and standard deviation (only needed when `distribution_type = "Normal"`); see `une689_statistics()`.
`MG, DSG`	Numeric. Geometric mean and standard deviation (only needed when `distribution_type = "Lognormal"`); see `une689_statistics()`.

Value

Numeric scalar with the LSC(95,70), or `NA_real_` if `distribution_type = "Neither"` (or `"Ninguna"` in Spanish).

Examples

```
une689_lsc("Normal", ut = 2.005, MA = 7.5, DS = 1.87)
```

---

une689_monitoring_interval_opt1

Recommended monitoring interval, option 1 (MG or MA vs VLA)

Description

Recommended monitoring interval, option 1 (MG or MA vs VLA)

Usage

```
une689_monitoring_interval_opt1(reference_value, vla)
```

Arguments

`reference_value`	Numeric. MG if the distribution is lognormal, or MA if normal (see `une689_statistics()`).
`vla`	Numeric. Valor Limite Ambiental / occupational exposure limit (mg/m3).

Value

Character scalar describing the recommended monitoring interval, in months.

Examples

```
une689_monitoring_interval_opt1(reference_value = 0.8, vla = 10)
```

une689_monitoring_interval_opt2 *Recommended monitoring interval, option 2 (LSC95,70 vs VLA)*

Description

Recommended monitoring interval, option 2 (LSC95,70 vs VLA)

Usage

```
une689_monitoring_interval_opt2(lsc, vla)
```

Arguments

lsc	Numeric. LSC(95,70), see une689_lsc().
vla	Numeric. Valor Limite Ambiental / occupational exposure limit (mg/m3).

Value

Character scalar describing the recommended monitoring interval, in months, or a warning that exposure must be reviewed.

Examples

```
une689_monitoring_interval_opt2(lsc = 4, vla = 10)
```

une689_normality_test *Normality and lognormality tests (UNE-EN 689)*

Description

Applies the Shapiro-Wilk test to the ED values (to test normality) and to their logarithm (to test lognormality).

Usage

```
une689_normality_test(ed)
```

Arguments

ed	Numeric vector. Daily exposure (ED) values, all strictly positive. At least 3 values are required (minimum required by stats::shapiro.test()); UNE-EN 689 additionally requires a minimum of 6 for the full statistical assessment.

Value

A list with elements `W_normal`, `pval_normal`, `W_lognormal`, `pval_lognormal`.

Examples

```
une689_normality_test(c(5, 6, 7, 8, 9, 10))
```

`une689_probability_data` *Probability-paper data for examining a similar exposure group (UNE-EN 689)*

Description

Computes the quantities needed to draw the graphical validation of a similar exposure group (SEG, "GES" in Spanish) on probability paper: the daily exposure values are ranked in ascending order and each rank `k` is assigned the plotting position $P_k = (k - 3/8)/(n + 1/4)$. If the data follow the assumed distribution, the points fall approximately on a straight line when plotted against P_k on log-normal paper (lognormal data) or on normal paper (normal data).

Usage

```
une689_probability_data(
  ed,
  distribution = c("auto", "lognormal", "normal"),
  alpha = 0.05
)
```

Arguments

`ed`	Numeric vector. Daily exposure (ED) values, all strictly positive; between 3 and 5000 values.
`distribution`	Character. `"lognormal"`, `"normal"`, or `"auto"` (default). With `"auto"` the log-normal distribution is used if the Shapiro-Wilk test on `log(ED)` does not reject it at level `alpha`; otherwise the normal distribution is used if the test on `ED` does not reject it; if both are rejected, an error is raised. The labels returned by `une689_distribution_type()` (English or Spanish) are also accepted.
`alpha`	Numeric. Significance level of the Shapiro-Wilk tests used by `distribution = "auto"` (default 0.05).

Details

The vertical coordinate used for the fit is the standard normal quantile $z_k = \Phi^{-1}(P_k)$, which is what the probability scale of the paper represents. The straight line is the least-squares regression of `log(ED)` (lognormal) or `ED` (normal) on z_k. For lognormal data the intercept and slope estimate `log(MG)` and `log(DSG)`, and for normal data they estimate `MA` and `DS`; these are regression-based graphical estimates and do not replace `une689_statistics()`.

The coefficient of determination returned in `r_squared` is purely descriptive. The graphical check is a visual one: the group is considered consistent when all points lie close to a straight line and none is exceptionally separated from the rest. This function does not apply any numerical cut-off to that judgement.

Note that `stats::ppoints()` uses the same plotting positions only for `n <= 10`; this function always uses $(k - 3/8)/(n + 1/4)$.

Value

A list with elements:

`data` A `data.frame` with one row per observation and columns `k` (rank), `ed` (sorted values), `pk` (plotting position), `pk_percent` (`100 * pk`), `z` (standard normal quantile of `pk`), `y` (`log(ed)` for lognormal, `ed` for normal), `fitted` and `residual` of the straight-line fit.

`distribution` `"lognormal"` or `"normal"`.

`n` Number of observations.

`intercept`, `slope` Coefficients of the straight line `y = intercept + slope * z`.

`r_squared` Coefficient of determination of the fit.

`W`, `p_value` Shapiro-Wilk statistic and p-value for the chosen distribution (on `log(ed)` for lognormal, on `ed` for normal).

`alpha` The significance level used.

See Also

`une689_probability_plot()`, `une689_normality_test()`, `une689_distribution_type()`

Examples

```
ed <- c(0.008, 0.012, 0.017, 0.021, 0.034, 0.041)
pd <- une689_probability_data(ed)
pd$distribution
pd$data[, c("k", "ed", "pk")]
pd$r_squared
```

`une689_probability_plot` *Probability paper plot for examining a similar exposure group (UNE-EN 689)*

Description

Draws the daily exposure (ED) values of a similar exposure group on probability paper: the vertical axis is the cumulative probability P_k on a normal-probability scale and the horizontal axis is the exposure on a logarithmic scale (log-normal paper) when the data are lognormal, or on a linear scale (normal paper) when the data are normal. A straight line is fitted to the points. If all points lie close to the line and none is exceptionally separated from the others, the group is consistent with the assumed distribution.

Usage

```
une689_probability_plot(
  ed,
  distribution = c("auto", "lognormal", "normal"),
  alpha = 0.05,
  vla = NULL,
  title = NULL,
  subtitle = NULL,
  x_label = "Daily exposure ED (mg/m³)",
  y_label = "Cumulative probability P_k (%)",
  vla_label = "OEL",
  digits = 3
)
```

Arguments

`ed`	Numeric vector. Daily exposure (ED) values, all strictly positive; between 3 and 5000 values.
`distribution`	Character. `"lognormal"`, `"normal"`, or `"auto"` (default). With `"auto"` the log-normal distribution is used if the Shapiro-Wilk test on `log(ED)` does not reject it at level `alpha`; otherwise the normal distribution is used if the test on `ED` does not reject it; if both are rejected, an error is raised. The labels returned by `une689_distribution_type()` (English or Spanish) are also accepted.
`alpha`	Numeric. Significance level of the Shapiro-Wilk tests used by `distribution = "auto"` (default 0.05).
`vla`	Optional positive numeric. Occupational exposure limit (mg/m3, expressed as a daily exposure). If supplied, it is drawn as a vertical reference line.
`title, subtitle`	Optional character scalars. If `NULL` (default) an informative title and subtitle are generated. Use `""` to suppress the subtitle.
`x_label, y_label`	Character scalars. Axis labels.

vla_label	Character scalar. Label of the OEL reference line.
digits	Non-negative integer. Number of decimal places used in the OEL annotation.

Details

The default subtitle reports the number of observations, the Shapiro-Wilk statistic and p-value for the chosen distribution, and the coefficient of determination of the line. See une689_probability_data() for how the plotting positions and the line are computed.

Requires the 'ggplot2' package.

Value

A ggplot object.

See Also

une689_probability_data()

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  ed <- c(0.008, 0.012, 0.017, 0.021, 0.034, 0.041)
  une689_probability_plot(ed, vla = 0.1)
}
```

---

une689_statistical_conformity

Conformity of the statistical assessment (UNE-EN 689)

Description

Conformity of the statistical assessment (UNE-EN 689)

Usage

```
une689_statistical_conformity(ur, ut)
```

Arguments

ur	Numeric. One-sided risk index, see une689_ur().
ut	Numeric. UT factor, see une689_ut().

Value

Character scalar: "CONFORMITY" if ur >= ut, "NON-CONFORMITY" if ur < ut, or NA_character_ if ur is NA (e.g. when neither the normal nor the lognormal fit is adequate). Labels are returned in the active language; see expoquimr_lang().

Examples

```
une689_statistical_conformity(ur = 2.1, ut = 2.005)
```

`une689_statistics` *Descriptive statistics of daily exposure (UNE-EN 689)*

Description

Calculates the arithmetic mean and standard deviation (MA, DS) and the geometric mean and standard deviation (MG, DSG) of a set of daily exposure (ED) values, needed to test the normal and lognormal fits.

Usage

```
une689_statistics(ed)
```

Arguments

`ed`	Numeric vector. Daily exposure (ED) values, all strictly positive.

Value

A list with elements `MA`, `DS`, `MG`, `DSG`.

Examples

```
une689_statistics(c(5, 6, 7, 8, 9, 10))
```

`une689_ur` *One-sided risk index UR (UNE-EN 689)*

Description

One-sided risk index UR (UNE-EN 689)

Usage

```
une689_ur(
  distribution_type,
  vla,
  MA = NA_real_,
  DS = NA_real_,
  MG = NA_real_,
  DSG = NA_real_
)
```

Arguments

distribution_type	Character. Distribution type; see une689_lsc().
vla	Numeric. Valor Limite Ambiental / occupational exposure limit (mg/m3).
MA, DS, MG, DSG	Numeric. See une689_lsc().

Value

Numeric scalar, or NA_real_ if distribution_type = "Neither".

Examples

```
une689_ur("Normal", vla = 10, MA = 7.5, DS = 1.87)
```

---

une689_ut *UNE-EN 689 UT factor from the sample size*

Description

Looks up the one-sided tolerance factor (UT) tabulated by UNE-EN 689 for a number of days n between 6 and 30. For n > 30 the limit value 1.820 is used, as established by the standard.

Usage

```
une689_ut(n)
```

Arguments

n	Integer. Number of days (ED measurements) used in the statistical assessment. Must be >= 6.

Value

Numeric scalar with the UT value, or NA_real_ if n < 6.

Examples

```
une689_ut(6)
une689_ut(50)
```

une689_validate_min_days *Check the minimum number of days for the preliminary assessment*

Description

The UNE-EN 689 preliminary assessment requires a minimum number of evaluated days (usually 3). Helper function to validate this before calculating, both from code and from the Shiny app.

Usage

```
une689_validate_min_days(n_days, minimum = 3L)
```

Arguments

n_days	Integer. Number of days with data entered.
minimum	Integer. Minimum number required (default 3).

Value

Logical: TRUE if n_days >= minimum.

Examples

```
une689_validate_min_days(2)
une689_validate_min_days(3)
```

Index

coshh_classify_volatility, 3
coshh_classify_volatility(), *4*
coshh_evaluate, 4
coshh_evaluate(), *5*, *9*
coshh_from_excel, 5
coshh_from_excel(), *9*
coshh_grade, 6
coshh_grade(), *4*, *8*
coshh_measures, 7
coshh_measures(), *4*
coshh_risk, 7
coshh_risk(), *4*, *7*

expoquimr_lang, 8
expoquimr_lang(), *3*, *5*, *7*, *10*, *35*
expoquimr_report, 9

inrs_collective_protection, 10
inrs_collective_protection(), *12*, *15*
inrs_evaluate, 11
inrs_evaluate(), *9*, *13, 14*
inrs_frequency_class, 13
inrs_frequency_class(), *12*, *18*
inrs_from_excel, 13
inrs_from_excel(), *9*
inrs_hazard_class, 14
inrs_hazard_class(), *12*, *18*
inrs_inhalation_risk, 15
inrs_inhalation_risk(), *20*
inrs_liquid_volatility_graph, 16
inrs_liquid_volatility_graph(), *22*
inrs_liquid_volatility_pressure, 16
inrs_liquid_volatility_pressure(), *22*
inrs_oel_correction_factor, 17
inrs_oel_correction_factor(), *15*
inrs_potential_exposure_class, 17
inrs_potential_exposure_class(), *18*
inrs_potential_risk_class, 18
inrs_potential_risk_class(), *19*
inrs_potential_risk_score, 19
inrs_potential_risk_score(), *15*
inrs_process_type, 19
inrs_process_type(), *12*, *15*
inrs_quantity_class, 20
inrs_quantity_class(), *12*, *18*
inrs_risk_characterisation, 20
inrs_solid_dustiness, 21
inrs_solid_dustiness(), *12*, *22*
inrs_volatility_score, 21
inrs_volatility_score(), *15*

options(), *8*

run_coshh, 22
run_inrs, 23
run_une689, 23

shiny::runApp(), *22, 23*
stats::ppoints(), *33*
stats::shapiro.test(), *31*
system.file(), *5*

une689_classify_conformity, 24
une689_classify_conformity(), *26*
une689_daily_exposure, 25
une689_daily_exposure(), *28*
une689_distribution_type, 25
une689_distribution_type(), *29*, *32–34*
une689_evaluate_preliminary, 26
une689_evaluate_statistical, 27
une689_exposure_index, 28
une689_exposure_index(), *24*
une689_from_excel, 28
une689_from_excel(), *9*
une689_lsc, 29
une689_lsc(), *31*, *37*
une689_monitoring_interval_opt1, 30
une689_monitoring_interval_opt2, 31
une689_normality_test, 31
une689_normality_test(), *33*

une689_probability_data, 32
une689_probability_data(), *35*
une689_probability_plot, 34
une689_probability_plot(), *33*
une689_statistical_conformity, 35
une689_statistics, 36
une689_statistics(), *29, 30*, *33*
une689_ur, 36
une689_ur(), *35*
une689_ut, 37
une689_ut(), *29*, *35*
une689_validate_min_days, 38
une689_validate_min_days(), *27*