

Package ‘rmoriedata’

September 17, 2026

Type Package

Title Integrated Datasets for the 'rmorie' Package

Version 0.3.2

Description Integrated open data fixtures used by the 'rmorie' package for examples, vignettes, and tests. Split out so 'rmorie' itself stays within the 'CRAN' package-size soft cap. Contains snapshots of publicly available datasets from open-data portals built on the Comprehensive Knowledge Archive Network ('CKAN', <<https://ckan.org/>>), 'Socrata' (<<https://dev.socrata.com/>>), and 'Opendatasoft' (<<https://www.huwise.com/>>) (Chicago, New York City, Toronto, Vancouver, and others), Statistics Canada Canadian Centre for Justice and Community Safety Statistics ('CCJS') tables, a multi-agent-reviewed corpus of Ontario Special Investigations Unit ('SIU', <<https://www.siu.on.ca/>>) director's reports, and synthetic fixtures for unit tests. Also ships a small set of analyst-facing helpers for releasing aggregate statistics without re-identification risk: Laplace and Gaussian differential privacy mechanisms and k-anonymity, l-diversity, and cell suppression verifiers.

License AGPL (>= 3)

Encoding UTF-8

Depends R (>= 4.3.0)

Imports stats, rmoriebricklayer (>= 0.2.1)

LinkingTo rmoriebricklayer

Suggests testthat (>= 3.0.0), nanoparquet, tibble, curl, jsonlite, knitr, rmarkdown

VignetteBuilder knitr

LazyData true

LazyDataCompression xz

Config/testthat/edition 3

URL <https://github.com/rootcoder007/rmoriedata>

BugReports <https://github.com/rootcoder007/rmoriedata/issues>

biocViews Software, ExperimentData
Config/roxygen2/version 8.0.0
RoxygenNote 7.3.3
NeedsCompilation yes
Author Vansh Singh Ruhela [aut, cre] (ORCID:
 <<https://orcid.org/0009-0004-1750-3592>>)
Maintainer Vansh Singh Ruhela <vsruhela@proton.me>
Repository CRAN
Date/Publication 2026-09-17 10:00:02 UTC

Contents

arrest_sample	2
ask	3
complaint_sample	4
fetch_cihi_table	6
load_chicago_data	7
load_cihi_data_tables	8
load_siu_reports	9
morie_cell_suppress	11
morie_core	12
morie_data_catalog	13
morie_data_checksums	14
morie_data_dictionary	15
morie_data_load	16
morie_dp_gaussian_mean	17
morie_dp_laplace_count	18
morie_dp_laplace_histogram	19
morie_k_anonymity_verify	20
morie_l_diversity_verify	21
rmoriedata-victoria	22
Index	24

arrest_sample	<i>Chicago arrests sample</i>
---------------	-------------------------------

Description

A CRAN-safe slice of the City of Chicago "Arrests" dataset. For the full ~1.5M-row dataset use `load_chicago_data("arrests", full = TRUE)`.

Usage

`arrest_sample`

Format

A base `data.frame` with up to 25,000 rows and 8 columns:

case_number Chicago PD records-division number (character).

date_iso Arrest timestamp as an ISO-8601 string with offset.

date Arrest timestamp as POSIXct (America/Chicago).

race Recorded race of the arrestee (character).

charge_type Charge type of the primary charge (F/M/etc.).

charge_class Charge class of the primary charge.

charge_desc Description of the primary charge.

charge_statute Statute of the primary charge.

Source

City of Chicago Open Data Portal, "Arrests" (dataset dpt3-jri9). <https://data.cityofchicago.org/>

See Also

[load_chicago_data](#)

Examples

```
data(arrest_sample)
dim(arrest_sample)
str(arrest_sample)

# Recorded race distribution.
sort(table(arrest_sample$race), decreasing = TRUE)

# Charge severity (F = felony, M = misdemeanour, ...).
sort(table(arrest_sample$charge_type), decreasing = TRUE)

# Most frequent primary charges.
head(sort(table(arrest_sample$charge_desc), decreasing = TRUE), 5)

# Cross-tab race x charge type.
with(arrest_sample, table(race, charge_type))
```

ask

Ask the rmore agent about the bundled datasets

Description

Convenience wrapper that forwards a dataset-focused question to the `rmore` command-line agent (optional binary from `rmore-cli`). See `rmore::agent` for the full interface and requirements.

Usage

```
ask(question, model = NULL, backend = "auto")
```

Arguments

question	Character scalar.
model	Optional model id (see <code>rmorie::agent</code>).
backend	Optional backend override (see <code>rmorie::agent</code>).

Value

Character scalar: the agent's output, or a message if the `rmorie` binary is not installed.

Examples

```
# Routed to the optional rmorie CLI agent when it is installed; with no
# binary on PATH each call returns an install hint instantly (no error,
# no network), so this is safe to execute anywhere.
# Plain question -> routed to the rmorie CLI agent (auto backend).
ask("which bundled datasets cover Toronto police use-of-force?")

# Pin a specific model.
ask("summarise the SIU director's-report corpus", model = "gpt-4o-mini")

# Force a backend (see rmorie::agent for the available values).
ask("list the Chicago datasets", backend = "ollama")

# With no rmorie binary on PATH the call returns an install hint, not an
# error -- safe to run anywhere:
if (!nzchar(Sys.which("rmorie"))) ask("hello")
```

complaint_sample	<i>Chicago reported-crime sample ("complaints")</i>
------------------	---

Description

A CRAN-safe slice of the City of Chicago "Crimes – 2001 to present" dataset (reported incidents), filtered to geocoded rows from 2020 onward. For the full dataset use `load_chicago_data("complaints", full = TRUE)`.

Usage

```
complaint_sample
```

Format

A base data.frame with up to 25,000 rows and 16 columns:

case_number Chicago PD records-division number (character).

date_iso Incident timestamp as an ISO-8601 string with offset (lossless across R/Python).

date Incident timestamp as POSIXct (America/Chicago).

iucr Illinois Uniform Crime Reporting code (character).

primary_type Primary FBI crime classification.

description Secondary description of the offense.

arrest Whether an arrest was made (logical).

domestic Whether domestic-violence related (logical).

beat,district,ward,community_area Geographic area codes (integer).

fbi_code FBI crime code (character).

year Year of the incident (integer).

latitude,longitude WGS84 coordinates (numeric).

Source

City of Chicago Open Data Portal, "Crimes - 2001 to present" (dataset ijzp-q8t2). <https://data.cityofchicago.org/>

See Also

[load_chicago_data](#)

Examples

```
data(complaint_sample)
dim(complaint_sample)
str(complaint_sample)

# Most common offense types.
head(sort(table(complaint_sample$primary_type), decreasing = TRUE), 5)

# Arrest rate among reported incidents.
mean(complaint_sample$arrest)

# Incidents per year (the sample spans 2020+).
table(complaint_sample$year)

# Domestic-violence-flagged incidents by type.
head(sort(table(complaint_sample$primary_type[complaint_sample$domestic]),
            decreasing = TRUE), 3)
```

fetch_cihi_table	<i>Download a CIHI data table (live, with Wayback fallback)</i>
------------------	---

Description

Resolves a CIHI table from the bundled catalogue and downloads it to `dest`, falling back to the Internet Archive snapshot if the live CIHI URL has rotated or been removed. The download + fallback runs through **rmoriebricklayer**'s shared C++/libcurl foundation (`bricklayer_fetch()`), the same engine **rmorie** and **morie** use – one implementation across the ecosystem.

Usage

```
fetch_cihi_table(which, dest = NULL, timeout = 120L)
```

Arguments

<code>which</code>	A row index into <code>load_cihi_data_tables()</code> , or a string matched (case-insensitively, as a substring) against table titles. Must resolve to exactly one table.
<code>dest</code>	Destination file path. Default: a tempfile with the table's own extension.
<code>timeout</code>	Per-request timeout, seconds.

Value

The `dest` path, invisibly. Errors if both the live URL and its Wayback fallback fail.

Examples

```
# Offline: inspect the catalogue to choose a `which` argument.
cat <- load_cihi_data_tables()
head(cat$title, 3)

# Downloads a table from the live CIHI web service; try() keeps the
# example graceful when the service is unreachable.
# `which` by title substring (case-insensitive; must match exactly one).
f1 <- try(fetch_cihi_table("Hospital Beds"))      # -> tempfile path

# `which` by row index into load_cihi_data_tables(); `dest` chooses the
# output path and `timeout` bounds each request (seconds).
f3 <- try(fetch_cihi_table(1, dest = tempfile(fileext = ".xlsx"),
                          timeout = 60))

# An ambiguous substring errors and lists the candidates:
try(fetch_cihi_table("data"))
```

load_chicago_data	<i>Load Chicago crime or arrest data</i>
-------------------	--

Description

Returns the bundled sample by default, or fetches the full dataset from the City of Chicago SODA API (cached under [R_user_dir](#)) when `full = TRUE`. The result can be returned as a base data frame, a tibble, or written to a Parquet file whose path is returned – the last being the recommended bridge for Python (`pandas.read_parquet`).

Usage

```
load_chicago_data(
  type = c("arrests", "complaints"),
  as = c("data.frame", "tibble", "parquet_path"),
  full = FALSE,
  mirror = getOption("rmoriedata.mirror", NULL),
  limit = NULL,
  fraction = NULL
)
```

Arguments

<code>type</code>	One of "arrests" or "complaints".
<code>as</code>	Return format: "data.frame" (default), "tibble", or "parquet_path" (writes a Parquet file to the session cache and returns its path).
<code>full</code>	If TRUE, fetch the complete dataset from Socrata (network, large) instead of the bundled sample; cached across sessions as Parquet.
<code>mirror</code>	Optional base URL of an r-universe/drat mirror to try before Socrata (offline-friendly fallback). Defaults to <code>getOption("rmoriedata.mirror")</code> .
<code>limit</code>	Optional row cap for a <code>full = TRUE</code> fetch (passed to the Socrata <code>\$limit</code> parameter). A bounded fetch skips the mirror and is never written to the full-dataset cache. Default NULL fetches everything.
<code>fraction</code>	Optional share of the dataset, in $(0, 1]$, for a <code>full = TRUE</code> fetch: the live row count is looked up and <code>limit</code> is set to <code>ceiling(total * fraction)</code> . Give either fraction or limit, not both.

Details

Parquet I/O uses **nanoparquet** (already a hard dependency of this package), so no **arrow** install is required.

Value

A `data.frame`/tibble, or a length-1 character Parquet path when `as = "parquet_path"`.

Examples

```
# `type` selects the dataset; the bundled sample is returned by default.
comp <- load_chicago_data("complaints")      # reported incidents
arr  <- load_chicago_data("arrests")        # arrests
nrow(comp); nrow(arr)
head(sort(table(comp$primary_type), decreasing = TRUE), 5)

# `as = "tibble"` returns a tibble when the package is installed.
if (requireNamespace("tibble", quietly = TRUE)) {
  tb <- load_chicago_data("complaints", as = "tibble")
  class(tb)
}

# `as = "parquet_path"` writes a Parquet file and returns its path --
# the recommended bridge to Python (pandas.read_parquet). Offline: the
# bundled sample is written, no network.
pq <- load_chicago_data("arrests", as = "parquet_path")
file.exists(pq)

# `full = TRUE` fetches from the live Chicago SODA API; `limit` bounds
# the request (seconds, not minutes) and try() keeps the example
# graceful when the service is unreachable. Omit `limit` for the
# complete multi-million-row dataset (cached across sessions); `mirror`
# tries an offline-friendly Parquet mirror first when set.
big <- try(load_chicago_data("complaints", full = TRUE, limit = 1000))
if (!inherits(big, "try-error")) nrow(big)

# `fraction` takes a share of the dataset instead of a row count:
# 0.001 = 0.1% of all rows (the live total is looked up first).
tiny <- try(load_chicago_data("arrests", full = TRUE, fraction = 0.0001))
if (!inherits(tiny, "try-error")) nrow(tiny)
```

load_cihi_data_tables *Catalogue of CIHI open data-table workbooks (with Wayback fallbacks)*

Description

Returns the bundled catalogue of the public data-table .xlsx workbooks published on the Canadian Institute for Health Information (CIHI) “Access data and reports > Data tables” page (<https://www.cihi.ca/en/access-data-and-reports/data-tables>). Each row carries the table title, its direct url, and a wayback_url snapshot on the Internet Archive so the table stays retrievable even if CIHI rotates or removes the live file.

Usage

```
load_cihi_data_tables(archived_only = FALSE)
```

Arguments

`archived_only` If TRUE, drop rows with no Wayback snapshot. Default FALSE (return the full catalogue).

Details

Pair with `rmorie::morie_ingest_cihi_xlsx()` to download + parse any row (that helper tries url first and falls back to `wayback_url`). The Wayback snapshots were resolved with `rmoriebricklayer::wayback_snapsh`

Value

A data.frame with columns `title`, `url`, `wayback_url`.

Source

Canadian Institute for Health Information, Data tables (<https://www.cihi.ca/en/access-data-and-reports/data-tables>). Snapshot to the Internet Archive (<https://web.archive.org>). Catalogue current as of 2026-07.

Examples

```
# Full catalogue: title, live url, Wayback snapshot url.
cat <- load_cihi_data_tables()
nrow(cat)
names(cat)
head(cat$title, 3)

# `archived_only = TRUE` keeps only rows that have a Wayback snapshot,
# i.e. tables still retrievable if CIHI rotates the live file.
arch <- load_cihi_data_tables(archived_only = TRUE)
nrow(arch)          # <= nrow(cat)
all(nzchar(arch$wayback_url)) # TRUE

# Find a table by keyword before fetching it.
cat$title[grepl("hospital", cat$title, ignore.case = TRUE)][1:3]
```

load_siu_reports

Load the Ontario SIU director's-report corpus

Description

Returns the bundled Ontario Special Investigations Unit (SIU) director's-report table: one row per report drid, 65 structured columns (police service, incident / notification / decision dates, investigator and witness / subject-official counts, affected-person demographics, injuries, legislation, charges verdict, director's decision, and news-release linkage), plus a `panel_reviewed` flag.

Usage

```
load_siu_reports(
  lang = c("all", "en", "fr"),
  as = c("data.frame", "tibble"),
  format = c("csv", "parquet")
)
```

Arguments

lang	One of "all" (default), "en", or "fr": filter to the English-only, French-only, or all rows.
as	Return format: "data.frame" (default) or "tibble".
format	Bundle to read: "csv" (default, the gzip CSV) or "parquet" (columnar, via nanoparquet). Both hold the identical corpus.

Details

For every English report (`panel_reviewed == "TRUE"`), the 16 key columns were verified by a multi-agent LLM review panel against the full report text and the parser's guess resolved to the correct value; the subject-official count is filled for 100% of English reports (witness-officer-only investigations are a genuine 0). French reports carry the parser values. See the `siu` pipeline repo for the audit provenance.

This is the machine-readable companion to the SIU parser and data-mining subsystem in **rmorie** / **morie** – the first open-source pipeline for the SIU director's-report corpus, created by Vansh Singh Ruhela as part of the MORIE / MRM framework. The table is regenerated from the parser over the full public corpus; see `rmorie::morie_fetch_siu()` to rebuild it live.

Value

A `data.frame` (or `tibble`) of SIU director's-report rows.

Source

Ontario Special Investigations Unit director's reports, https://www.siu.on.ca/en/directors_reports.php (post-2018) and the Ontario Government archive (pre-2018). Parsed with the **rmorie** SIU subsystem.

Examples

```
# Default: every parsed report, as a base data.frame.
all <- load_siu_reports()
nrow(all)
ncol(all)

# `lang` filters the corpus by report language.
en <- load_siu_reports(lang = "en") # English director's reports
fr <- load_siu_reports(lang = "fr") # French director's reports
nrow(en); nrow(fr)
```

```
# `as = "tibble"` returns a tibble when the tibble package is present.
if (requireNamespace("tibble", quietly = TRUE)) {
  tb <- load_siu_reports(lang = "en", as = "tibble")
  class(tb)
}

# The five police services with the most reports.
if (nrow(en)) {
  top <- sort(table(en$police_service), decreasing = TRUE)
  head(top, 5)
}
```

morie_cell_suppress *Cell suppression with optional complementary suppression*

Description

Standard StatCan / open-data complementary-suppression: identifies counts below ‘threshold’, suppresses them by setting to ‘NA’, and (if ‘return_complementary = TRUE’) also suppresses the smallest other count in each affected row and column so the suppressed value can’t be reconstructed from marginals.

Usage

```
morie_cell_suppress(tbl, threshold = 5, return_complementary = TRUE)
```

Arguments

tbl	A numeric matrix or 2-D table of counts. Will be coerced to matrix; row/column names are preserved.
threshold	Minimum count to remain unsuppressed. Default 5.
return_complementary	Logical; if TRUE (default), apply complementary suppression so primary-suppressed cells can’t be recovered from marginal sums.

Details

Only finite numeric cells are eligible for suppression. NA cells in the input pass through unchanged.

Value

A list with class "morie_cell_suppress":

suppressed numeric matrix, suppressed cells set to NA.

primary_mask logical matrix, TRUE for primary suppressions.

complementary_mask logical matrix, TRUE for complementary suppressions (all FALSE when return_complementary = FALSE).

n_primary integer.
 n_complementary integer.
 threshold the threshold used.

Examples

```
tbl <- matrix(c(120, 3, 47, 88, 2, 99, 14, 51, 60), nrow = 3,
              dimnames = list(c("A", "B", "C"), c("X", "Y", "Z")))

# Default: primary suppression (cells 1..4) PLUS complementary suppression
# so a suppressed cell can't be recovered from row/column marginals.
res <- morie_cell_suppress(tbl, threshold = 5)
res$suppressed          # NA where suppressed
res$n_primary           # cells below threshold
res$n_complementary      # extra cells hidden to protect the marginals
res$primary_mask

# Turn complementary suppression off: only the small cells are hidden.
morie_cell_suppress(tbl, threshold = 5,
                    return_complementary = FALSE)$suppressed

# A higher threshold suppresses more cells.
morie_cell_suppress(tbl, threshold = 50)$n_primary

# Works on a 2-D table too; NA cells pass through untouched.
t2 <- as.table(matrix(c(2, 40, 30, 1), 2,
                      dimnames = list(c("a", "b"), c("c", "d"))))
morie_cell_suppress(t2, threshold = 5)$suppressed
```

morie_core

Shared C-core helpers (rmorie ecosystem backend)

Description

Thin access to the compiled core that ships in **rmoriebricklayer**. `rmoriedata` links that core via ‘LinkingTo: rmoriebricklayer’, so these functions call the exact same kernels used across the `rmorie` family – no duplicated C code. They back fast data-integrity hashing and summaries for the bundled datasets without requiring **rmorie**.

Usage

```
morie_core_sha256(x)

morie_core_mean(x)
```

Arguments

`x` For ‘`morie_core_sha256()`’, a length-1 character vector or a raw vector. For ‘`morie_core_mean()`’, a numeric vector (coerced with `[as.numeric()]`); NA/NaN propagate.

Value

‘morie_core_sha256()’ returns a 64-character lowercase hex digest. ‘morie_core_mean()’ returns a length-1 numeric.

Examples

```
## ---- morie_core_sha256(): 64-char lowercase hex digest -----
morie_core_sha256("abc")          # hash a character scalar
morie_core_sha256("")             # the empty string still hashes
morie_core_sha256(charToRaw("abc")) # identical digest from raw bytes

# character input and its raw-byte equivalent agree:
identical(morie_core_sha256("abc"), morie_core_sha256(charToRaw("abc")))

# Data-integrity pin: verify a value is byte-for-byte what you expect.
expected <- morie_core_sha256("record-42")
stopifnot(morie_core_sha256("record-42") == expected)

# Fingerprint a whole object by hashing its serialization.
morie_core_sha256(serialize(list(a = 1, b = "x"), NULL))

## ---- morie_core_mean(): fast length-1 mean -----
morie_core_mean(1:10)             # 5.5
morie_core_mean(c(2, 4, 6))       # 4
morie_core_mean(c(-1, 0, 1))      # 0
morie_core_mean(c(1, 2, NA))      # NA propagates (no na.rm)
morie_core_mean(complaint_sample$year) # mean over a bundled column
```

morie_data_catalog *Catalogue of bundled datasets*

Description

Lists every dataset in the bundled Parquet store, including row/column counts and the original source path each table was built from.

Usage

```
morie_data_catalog()
```

Value

A ‘data.frame’ with columns ‘slug’, ‘source_path’, ‘kind’, ‘n_rows’, ‘n_cols’.

See Also

[morie_data_load()], [morie_data_dictionary()]

Examples

```

cat <- morie_data_catalog()
str(cat)

# How many datasets of each kind are bundled?
table(cat$kind)

# The tables, largest first.
tbls <- cat[cat$kind == "table", c("slug", "n_rows", "n_cols")]
head(tbls[order(-tbls$n_rows), ])

# Every slug you can pass to morie_data_load().
head(cat$slug, 10)

# Where each table was originally built from.
head(cat[, c("slug", "source_path")])

```

`morie_data_checksums` *SHA256 checksums of bundled rmoriedata files*

Description

Computes the SHA256 digest of every file rmoriedata bundles in `inst/extdata`, using the shared provenance layer ([sha256_file](#)). This lets an analysis verify it used the exact data slice rmoriedata shipped, and is rmoriedata's integration with the bricklayer provenance layer.

Usage

```
morie_data_checksums()
```

Value

A data frame with one row per bundled file and columns `file`, `bytes`, and `sha256`.

Examples

```

# One row per bundled file: name, size in bytes, SHA256 digest.
ck <- morie_data_checksums()
str(ck)
head(ck)

# Total bundled payload and the largest few files.
sum(ck$bytes)
head(ck[order(-ck$bytes), c("file", "bytes")], 3)

# Provenance workflow: pin the digest of a file you depend on, then
# assert it hasn't changed under you in a later session / reinstall.
if (nrow(ck)) {
  pinned <- ck$sha256[1]
}

```

```
again <- morie_data_checksums()
stopifnot(again$sha256[again$file == ck$file[1]] == pinned)
}
```

`morie_data_dictionary` *Data dictionary (JSON) for a dataset, if one is bundled*

Description

Data dictionary (JSON) for a dataset, if one is bundled

Usage

```
morie_data_dictionary(slug)
```

Arguments

`slug` Dictionary slug; see `[morie_data_catalog()]` rows where `'kind == "dictionary"'`.

Value

A character scalar of JSON, or `'NULL'` if no dictionary exists.

See Also

`[morie_data_catalog()]`

Examples

```
# Which dictionaries are bundled?
cat <- morie_data_catalog()
dict_slugs <- cat$slug[cat$kind == "dictionary"]
dict_slugs

# Fetch one dictionary's JSON (returns a character scalar of JSON).
if (length(dict_slugs)) {
  js <- morie_data_dictionary(dict_slugs[1])
  substr(js, 1, 200)
  # Parse it if you have jsonlite:
  if (requireNamespace("jsonlite", quietly = TRUE))
    str(jsonlite::fromJSON(js), max.level = 1)
}

# Unknown / non-dictionary slug: informative message, returns NULL.
morie_data_dictionary("no_such_dictionary")
```

<code>morie_data_load</code>	<i>Load a bundled dataset by slug</i>
------------------------------	---------------------------------------

Description

Load a bundled dataset by slug

Usage

```
morie_data_load(slug)
```

Arguments

`slug` Dataset slug; see the ‘slug’ column of `[morie_data_catalog()]`.

Value

A ‘data.frame’.

See Also

`[morie_data_catalog()]`

Examples

```
# Load a bundled lookup table by its slug.
iucr <- morie_data_load("chicago_iucr_codes")
str(iucr)
head(iucr)

# Any slug from the catalogue works the same way.
hoods <- morie_data_load("chicago_neighborhoods")
offense <- morie_data_load("nyc_nypd_offense_codes")
nrow(hoods); nrow(offense)

# Slugs are validated: an unknown one errors with guidance.
try(morie_data_load("no_such_dataset"))

# Pattern: pick a slug programmatically from the catalogue, then load it.
cat <- morie_data_catalog()
slug <- cat$slug[cat$kind == "table"][1]
head(morie_data_load(slug))
```

morie_dp_gaussian_mean

Differentially-private mean via the Gaussian mechanism with bounded inputs

Description

Releases an approximately (ϵ, δ) -DP mean of a bounded numeric vector. Sensitivity is derived from the user-asserted bounds: changing one record can shift the sum by at most upper - lower, so the mean's sensitivity is (upper - lower) / length(x).

Usage

```
morie_dp_gaussian_mean(x, lower, upper, epsilon, delta = 1e-06)
```

Arguments

x	Numeric vector (no NAs).
lower, upper	Hard bounds on 'x'. Caller must guarantee all(x >= lower & x <= upper); the function clips defensively but emits a warning if clipping was necessary.
epsilon, delta	Privacy parameters. Standard recommendation: delta < 1/length(x), epsilon in 0.1 to 5.0.

Details

The noise standard deviation follows the classical analytic-Gaussian calibration:

$$\sigma = \frac{\Delta \cdot \sqrt{2 \ln(1.25/\delta)}}{\epsilon}.$$

Value

A noised mean (single numeric).

Examples

```
set.seed(1)
x <- runif(1000, 0, 1)

# A private mean of bounded data (bounds asserted by the caller).
morie_dp_gaussian_mean(x, lower = 0, upper = 1, epsilon = 1.0)
mean(x)                                # the true mean, for comparison

# `delta` controls the (epsilon, delta) guarantee; smaller = stronger.
morie_dp_gaussian_mean(x, 0, 1, epsilon = 1.0, delta = 1e-9)

# Wider bounds raise sensitivity, so the same epsilon adds more noise.
morie_dp_gaussian_mean(x, lower = -5, upper = 5, epsilon = 1.0)
```

```
# Out-of-range values are clipped to [lower, upper] (with a warning).
y <- c(x, 2, -1)
suppressWarnings(morie_dp_gaussian_mean(y, lower = 0, upper = 1, epsilon = 1))
```

`morie_dp_laplace_count`

Differentially-private count via the Laplace mechanism

Description

Adds Laplace noise calibrated to sensitivity / epsilon. Use when releasing counts of records matching some predicate (e.g. number of UoF incidents in a division-year). Sensitivity is hardcoded to 1: one record entering or leaving the dataset changes the count by at most 1.

Usage

```
morie_dp_laplace_count(true_count, epsilon)
```

Arguments

<code>true_count</code>	Non-negative integer; the true count.
<code>epsilon</code>	Privacy budget (smaller = more noise = stronger privacy). Typical range: 0.1 to 5.0.

Details

Pure (ϵ , 0)-differentially-private under the standard add-or-remove-one neighbouring-databases definition.

Value

A noised count (numeric, may be fractional or negative). Caller should usually clip to a non-negative integer for display: `round(pmax(0, x))`.

Examples

```
set.seed(1)
# A single noised release of a true count of 42.
morie_dp_laplace_count(true_count = 42, epsilon = 1.0)

# Smaller epsilon = stronger privacy = more noise.
morie_dp_laplace_count(42, epsilon = 0.1) # noisier
morie_dp_laplace_count(42, epsilon = 5.0) # closer to 42

# The mechanism is unbiased: averaging many releases returns ~the truth.
mean(replicate(2000, morie_dp_laplace_count(42, epsilon = 1.0)))

# For display, clip to a non-negative integer.
round(pmax(0, morie_dp_laplace_count(3, epsilon = 0.5)))
```

morie_dp_laplace_histogram

Differentially-private histogram via the Laplace mechanism

Description

Adds independent Laplace($1/\epsilon$) noise to each bin count. Under the add-or-remove-one neighbouring-databases definition a single record participates in exactly one bin, so the per-bin sensitivity is 1 and the overall mechanism is $(\epsilon, 0)$ -DP.

Usage

```
morie_dp_laplace_histogram(counts, epsilon)
```

Arguments

counts	Integer vector of non-negative bin counts.
epsilon	Privacy budget (positive scalar).

Value

A numeric vector of the same length as ‘counts’. May contain fractional or negative values. Caller is responsible for any post-hoc non-negativity / rounding before display.

Examples

```
set.seed(1)
true <- c(120, 45, 8, 230, 17)

# Independent Laplace noise added to every bin.
morie_dp_laplace_histogram(true, epsilon = 0.5)

# Smaller epsilon = more noise per bin.
morie_dp_laplace_histogram(true, epsilon = 0.1)

# Post-process for display: clip negatives, round to integers.
noisy <- morie_dp_laplace_histogram(true, epsilon = 1.0)
round(pmax(0, noisy))

# Release a private histogram straight from tabulated data.
counts <- as.integer(table(complaint_sample$year))
morie_dp_laplace_histogram(counts, epsilon = 1.0)
```

```
morie_k_anonymity_verify
      k-anonymity verification
```

Description

Checks whether a data.frame satisfies k-anonymity over the supplied quasi-identifier columns. A dataset is k-anonymous if every combination of quasi-identifier values appears in at least ‘k’ rows.

Usage

```
morie_k_anonymity_verify(data, quasi_identifiers, k = 5)
```

Arguments

data	data.frame.
quasi_identifiers	Character vector of column names.
k	Minimum equivalence-class size. Default 5 (a common public-health / open-data threshold).

Value

A list with class "morie_k_anon" containing:

- satisfies logical, whether the dataset is k-anonymous.
- k the threshold used.
- min_class_size integer, size of the smallest class.
- n_classes integer, total number of equivalence classes.
- n_violations integer, number of classes below the threshold.
- violating_classes data.frame of class keys plus their .n sizes (empty data.frame when none).
- summary human-readable one-line summary.

Examples

```
df <- data.frame(
  age = c(25, 25, 25, 32, 32, 40),
  sex = c("F", "F", "F", "M", "M", "M")
)

# k = 2: the class {age=40, sex=M} has only 1 row -> VIOLATED.
res <- morie_k_anonymity_verify(df, c("age", "sex"), k = 2)
res$summary
res$satisfies
res$violating_classes      # the offending quasi-identifier combos
```

```
# Loosening to k = 1 always holds; the default k = 5 is stricter.
morie_k_anonymity_verify(df, c("age", "sex"), k = 1)$satisfies
morie_k_anonymity_verify(df, c("age", "sex"))$satisfies    # k = 5

# A single quasi-identifier is fine too.
morie_k_anonymity_verify(df, "sex", k = 3)$min_class_size

# On real bundled data: are (year, arrest) cells 5-anonymous?
morie_k_anonymity_verify(complaint_sample,
  c("year", "arrest"), k = 5)$summary
```

```
morie_l_diversity_verify
      l-diversity verification
```

Description

Checks whether a data.frame satisfies l-diversity: within each equivalence class defined by the quasi-identifiers, the sensitive attribute must take at least ‘l’ distinct values.

Usage

```
morie_l_diversity_verify(data, quasi_identifiers, sensitive, l = 3)
```

Arguments

data	data.frame.
quasi_identifiers	Character vector of QI column names.
sensitive	Name of the sensitive-attribute column.
l	Minimum number of distinct sensitive values per class. Default 3.

Value

A list with class "morie_l_div" containing:

- satisfies logical.
- l the threshold used.
- min_diversity integer, lowest per-class distinct count.
- n_classes integer.
- n_violations integer, classes below the threshold.
- violating_classes data.frame of class keys plus their .diversity count.
- summary human-readable.

Examples

```
df <- data.frame(
  age = c(25, 25, 25, 25, 32, 32, 32),
  sex = c("F", "F", "F", "F", "M", "M", "M"),
  dx = c("A", "B", "C", "A", "X", "Y", "Z")
)

# Class {25,F} has 3 distinct dx (A,B,C); {32,M} has 3 (X,Y,Z) -> l=3 holds.
res <- morie_l_diversity_verify(df, c("age", "sex"), "dx", l = 3)
res$summary
res$satisfies
res$min_diversity

# Demanding l = 4 fails: no class has 4 distinct sensitive values.
bad <- morie_l_diversity_verify(df, c("age", "sex"), "dx", l = 4)
bad$satisfies
bad$violating_classes

# k-anonymity and l-diversity are complementary: check both.
morie_k_anonymity_verify(df, c("age", "sex"), k = 3)$satisfies
```

rmoriedata-victoria *Victorian crime statistics (Crime Statistics Agency Victoria)*

Description

Ten tables from the Crime Statistics Agency's "Latest Victorian crime data" release, bundled in the Parquet store and reached by slug through `morie_data_load()`. Each is Table 01 – the headline series – of the corresponding published workbook, for the year ending March 2026.

Details

`vic_criminal_incidents` Criminal incidents by offence division, subdivision and subgroup, with rate per 100,000.

`vic_recorded_offences` Recorded offences on the same offence hierarchy.

`vic_victim_reports` Victim reports by offence.

`vic_alleged_offender_incidents` Alleged offender incidents, including age and sex breakdowns.

`vic_family_incidents` Family incidents by category and outcome.

`vic_lga_criminal_incidents`, `vic_lga_victim_reports`, `vic_lga_family_incidents` The same measures by police region and Local Government Area.

`vic_indigenous_victim_reports`, `vic_indigenous_family_incidents` Aboriginal and/or Torres Strait Islander status breakdowns, as published.

The workbooks are .xlsx. They were read with rmorie's native reader, so the bundled data comes through the same code path a user hits – no **readxl** or **openxlsx** dependency, and no second parser that could disagree with the first. Rebuild with `data-raw/build_vic_tables.R`.

Counts are as published by the CSA and are subject to its own revisions: figures for a given year change between releases as incidents are reclassified, so a table bundled here is a snapshot of the March 2026 release, not a permanent record of that year.

Source

Crime Statistics Agency Victoria, "Latest Victorian crime data". <https://www.crimestatistics.vic.gov.au/crime-statistics/latest-victorian-crime-data> Released under CC BY 4.0.

See Also

`morie_data_catalog()`, `morie_data_load()`

Examples

```
# Every bundled Victorian table, by slug.
cat <- morie_data_catalog()
cat[grepl("^vic_", cat$slug), c("slug", "n_rows", "n_cols")]

# Headline criminal-incident series.
ci <- morie_data_load("vic_criminal_incidents")
str(ci)

# Incidents by offence division for the most recent year.
latest <- ci[ci$Year == max(ci$Year), ]
tapply(latest[["Incidents Recorded"]], latest[["Offence Division"]], sum)
```

Index

* datasets

- arrest_sample, [2](#)
- complaint_sample, [4](#)
- rmoriedata-victoria, [22](#)

arrest_sample, [2](#)
ask, [3](#)

complaint_sample, [4](#)

fetch_cihi_table, [6](#)

load_chicago_data, [3](#), [5](#), [7](#)
load_cihi_data_tables, [8](#)
load_siu_reports, [9](#)

morie_cell_suppress, [11](#)
morie_core, [12](#)
morie_core_mean (morie_core), [12](#)
morie_core_sha256 (morie_core), [12](#)
morie_data_catalog, [13](#), [23](#)
morie_data_checksums, [14](#)
morie_data_dictionary, [15](#)
morie_data_load, [16](#), [22](#), [23](#)
morie_dp_gaussian_mean, [17](#)
morie_dp_laplace_count, [18](#)
morie_dp_laplace_histogram, [19](#)
morie_k_anonymity_verify, [20](#)
morie_l_diversity_verify, [21](#)

R_user_dir, [7](#)
rmoriedata-victoria, [22](#)

sha256_file, [14](#)