

# Package: **lmimpute** (via **r-universe**)

June 23, 2026

**Type** Package

**Title** Missing Data Imputation via Language Models and Statistics

**Version** 0.1.0

**Description** Provides missing data imputation through two complementary engines: a large language model engine that communicates with the 'Anthropic' 'Claude' application programming interface for context-aware semantic imputation, and a fully self-contained offline engine implementing nineteen statistical and machine learning algorithms entirely in base R with no additional package dependencies. Offline methods include mean, median, mode, last observation carried forward, next observation carried backward, hot-deck, predictive mean matching, k-nearest neighbours, ordinary least-squares regression, Lasso with coordinate descent, Ridge with closed-form solution, Bayesian Ridge regression with evidence approximation following MacKay (1992), support vector regression with a radial basis function kernel, classification and regression trees, random forests, gradient boosting, iterative random forest imputation, principal component analysis imputation via iterative singular value decomposition, and nuclear-norm minimisation via singular value thresholding. When no API key is available the package automatically falls back to the offline engine, ensuring full operation in environments without internet access. Every imputed value is accompanied by a confidence score and a plain-language reasoning string, producing reproducible audit trails. The automatic method selector chooses the best algorithm per column based on data type, skewness, missingness rate, and inter-column correlations.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Language** en-US

**Config/Needs/check** spelling

**Depends** R (>= 4.1.0)

**Imports** httr2 (>= 1.0.0), methods, jsonlite (>= 1.8.0), cli (>= 3.6.0),

**Suggests** testthat (>= 3.0.0), knitr (>= 1.40), rmarkdown (>= 2.14), withr (>= 2.5.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**BugReports** <https://cran.r-project.org/submit.html>

**NeedsCompilation** no

**Maintainer** Sadikul Islam <sadikul.islamiasri@gmail.com>

**Author** Sadikul Islam [aut, cre] (ORCID: <<https://orcid.org/0000-0003-2924-7122>>), Rajesh Kaushal [aut]

**Repository** <https://cran.r-universe.dev>

**Date/Publication** 2026-06-23 15:28:33 UTC

**RemoteUrl** <https://github.com/cran/lmimpute>

**RemoteRef** HEAD

**RemoteSha** a4214af4b5757f2ff4d1c4d07aee9ebbce8db6e3

## Contents

|                                    |    |
|------------------------------------|----|
| as.data.frame.lmi_result . . . . . | 2  |
| lmi_diagnose . . . . .             | 3  |
| lmi_export . . . . .               | 4  |
| lmi_impute . . . . .               | 5  |
| lmi_impute_offline . . . . .       | 7  |
| lmi_methods . . . . .              | 9  |
| lmi_models . . . . .               | 9  |
| lmi_providers . . . . .            | 10 |
| lmi_set_api_key . . . . .          | 10 |
| lmi_set_model . . . . .            | 12 |
| print.lmi_result . . . . .         | 13 |
| summary.lmi_result . . . . .       | 14 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>15</b> |
|--------------|-----------|

---

as.data.frame.lmi\_result

*Extract the imputed data frame from an lmi\_result*

---

## Description

Convenience S3 method allowing `as.data.frame(result)` to extract the imputed data frame from an `lmi_result` object directly.

**Usage**

```
## S3 method for class 'lmi_result'  
as.data.frame(x, ...)
```

**Arguments**

x                    An object of class `lmi_result`.  
...                  Currently unused. Included for S3 compatibility.

**Value**

The imputed data frame.

**Examples**

```
df <- data.frame(x = c(1, NA, 3))  
result <- lmi_impute_offline(df, verbose = FALSE)  
clean <- as.data.frame(result)
```

---

|                           |  |
|---------------------------|--|
| <code>lmi_diagnose</code> | <i>Diagnose missing data in a data frame</i> |
|---------------------------|--|

---

**Description**

Analyses a data frame and prints a report on missing values, column types, and the number of unique observed values per column. No API calls are made. Use this function before [lmi\\_impute](#) to preview what will be imputed and to choose an appropriate method.

**Usage**

```
lmi_diagnose(data, na_strings = NULL)
```

**Arguments**

data                A data frame or tibble.  
na\_strings          Character vector of additional strings to treat as NA (e.g. `c("N/A", "?", "-")`).  
Default NULL.

**Value**

Invisibly returns a data frame with one row per column and columns `column`, `type`, `n_missing`, `pct_missing`, `n_unique`.

**See Also**

[lmi\\_impute](#), [lmi\\_impute\\_offline](#)

**Examples**

```
df <- data.frame(
  age    = c(25L, NA, 35L, 40L),
  income = c(50000, 60000, NA, 80000),
  edu    = c("BSc", NA, "MSc", "BSc"),
  stringsAsFactors = FALSE
)
lmi_diagnose(df)
lmi_diagnose(df, na_strings = "N/A")
```

---

lmi\_export

*Export imputed data and audit trail to files*


---

**Description**

Writes the imputed data frame and the imputation audit trail to disk. Supports CSV (default) and RDS output formats. When `flag_suspicious = TRUE` was used in `lmi_impute`, the suspicious-values table is also written.

**Usage**

```
lmi_export(
  result,
  path = tempdir(),
  prefix = "l1mimpute",
  format = c("csv", "rds"),
  overwrite = FALSE
)
```

**Arguments**

|           |   |
|-----------|---|
| result    | An object of class <code>lmi_result</code> returned by <code>lmi_impute</code> or <code>lmi_impute_offline</code> .                       |
| path      | Character string. Output directory. Created recursively if it does not exist. Default is the current working directory <code>"."</code> . |
| prefix    | Character string prepended to output file names. Default <code>"l1mimpute"</code> .   |
| format    | Character string. <code>"csv"</code> (default) or <code>"rds"</code> .  |
| overwrite | Logical. Overwrite existing files? Default <code>FALSE</code> .   |

**Value**

Invisibly returns a named character vector of the file paths written.

**See Also**

[lmi\\_impute](#), [lmi\\_impute\\_offline](#)

## Examples

```
df <- data.frame(
  age = c(25L, NA, 35L),
  income = c(50000, 60000, NA),
  stringsAsFactors = FALSE
)
result <- lmi_impute_offline(df, verbose = FALSE)
## Not run:
lmi_export(result, path = tempdir(), prefix = "my_study")

## End(Not run)
```

---

lmi\_impute

*Impute missing values using LLM or built-in statistical methods*

---

## Description

Primary entry point for the **llmimpute** package. When an Anthropic API key is configured, missing values are filled using the Claude large language model, which reasons about each missing cell using the semantic meaning of column names, inter-column relationships, and domain knowledge. When no API key is available (or when `offline = TRUE`), the function transparently delegates to [lmi\\_impute\\_offline](#), which runs entirely in base R without internet access.

## Usage

```
lmi_impute(
  data,
  domain = c("general", "healthcare", "financial", "hr", "survey", "scientific"),
  offline = FALSE,
  offline_fallback = TRUE,
  offline_method = c("auto", "mean", "median", "mode", "locf", "nocb", "hotdeck", "pmm",
    "knn", "linear", "lasso", "ridge", "bayesian_ridge", "svr", "decision_tree",
    "random_forest", "gradient_boost", "missforest", "pca_impute", "softimpute"),
  na_strings = NULL,
  cols = NULL,
  confidence = TRUE,
  reasoning = TRUE,
  flag_suspicious = FALSE,
  max_rows = 50L,
  knn_k = 5L,
  seed = 42L,
  verbose = TRUE
)
```

**Arguments**

|                  |   |
|------------------|---|
| data             | A data.frame or tibble containing the data to impute.   |
| domain           | Character string describing the data domain. Guides LLM reasoning. One of "general" (default), "healthcare", "financial", "hr", "survey", "scientific". Ignored in offline mode.  |
| offline          | Logical. If TRUE, always use the offline statistical engine regardless of API key availability. Default FALSE.  |
| offline_fallback | Logical. If TRUE (default), switch to the offline engine when no API key is found instead of raising an error.  |
| offline_method   | Character. Offline imputation strategy passed to <code>lmi_impute_offline</code> . One of "auto" (default), "mean", "median", "mode", "locf", "nocb", "hotdeck", "pmm", "knn", "linear", "lasso", "ridge", "bayesian_lasso", "svr", "decision_tree", "random_forest", "gradient_boost", "missforest", "pca_impute", "softimpute". |
| na_strings       | Character vector of additional strings to treat as NA, e.g. <code>c("N/A", "?", "-")</code> . Default NULL.   |
| cols             | Character vector of column names to impute. NULL imputes all columns with any missing values.   |
| confidence       | Logical. Include a confidence score (0-100) per imputed cell. Default TRUE.   |
| reasoning        | Logical. Include a one-sentence explanation per imputed cell. Default TRUE.   |
| flag_suspicious  | Logical. Ask the LLM to flag anomalous existing values (LLM mode only). Default FALSE.  |
| max_rows         | Integer. Maximum rows per API call chunk (LLM mode only). Default 50L.  |
| knn_k            | Integer. Neighbours for the "knn" offline method. Default 5L.   |
| seed             | Integer. Random seed for reproducible offline imputation. Default 42L.  |
| verbose          | Logical. Print progress messages. Default TRUE.   |

**Value**

An object of class `lmi_result`, a named list with:

|             |   |
|-------------|---|
| data        | The imputed data.frame.   |
| imputations | data.frame audit trail: one row per imputed cell with columns row, col, original, imputed, confidence, reasoning. |
| suspicious  | data.frame of flagged existing values (LLM mode, <code>flag_suspicious = TRUE</code> only). NULL otherwise.       |
| summary     | Named list of imputation statistics.  |
| call        | The matched call.   |

**Engine selection**

**Situation**

API key present, offline = FALSE  
 No key, offline\_fallback = TRUE  
 No key, offline\_fallback = FALSE  
 offline = TRUE

**Behaviour**

LLM imputation (Anthropic)  
 Offline engine (silent)  
 Error with guidance  
 Offline engine always

**See Also**

[lmi\\_impute\\_offline](#), [lmi\\_diagnose](#), [lmi\\_export](#), [print.lmi\\_result](#)

**Examples**

```
# Offline imputation (works with no API key)
df <- data.frame(
  age    = c(25L, NA, 35L, 40L, NA),
  income = c(50000, 60000, NA, 80000, 55000),
  edu    = c("BSc", NA, "MSc", "BSc", "PhD"),
  stringsAsFactors = FALSE
)

result <- lmi_impute(df)
result$data
result$imputations
summary(result)

# Force a specific offline method
result2 <- lmi_impute(df, offline = TRUE, offline_method = "random_forest")

## Not run:
# LLM mode requires a valid Anthropic API key
lmi_set_api_key() # reads ANTHROPIC_API_KEY from environment
result3 <- lmi_impute(df, domain = "hr")

## End(Not run)
```

---

`lmi_impute_offline`     *Impute missing values using built-in statistical and ML methods (no API required)*

---

**Description**

Fully offline imputation using 20 methods from scratch in base R. No API key, no internet, no third-party packages needed.

**Usage**

```
lmi_impute_offline(
  data,
  method = c("auto", "mean", "median", "mode", "locf", "nocb", "hotdeck", "pmm", "knn",
    "linear", "lasso", "ridge", "bayesian_ridge", "svr", "decision_tree",
    "random_forest", "gradient_boost", "missforest", "pca_impute", "softimpute"),
  cols = NULL,
  na_strings = NULL,
  knn_k = 5L,
  n_trees = 50L,
  max_depth = 4L,
  shrinkage = 0.1,
  n_iter = 10L,
  lambda = 1,
  rank = 3L,
  seed = 42L,
  verbose = TRUE
)
```

**Arguments**

|            |   |
|------------|---|
| data       | A data.frame or tibble.   |
| method     | One of: "auto", "mean", "median", "mode", "locf", "nocb", "hotdeck", "pmm", "knn", "linear", "lasso", "ridge", "bayesian_ridge", "svr", "decision_tree", "random_forest", "gradient_boost", "missforest", "pca_impute", "softimpute". |
| cols       | Column names to impute. NULL = all missing columns.   |
| na_strings | Extra strings treated as NA.  |
| knn_k      | Neighbours for knn. Default 5L.   |
| n_trees    | Trees for rf/gb/missforest. Default 50L.  |
| max_depth  | Tree depth. Default 4L.   |
| shrinkage  | Learning rate for gradient_boost. Default 0.1.  |
| n_iter     | Iterations for iterative methods. Default 10L.  |
| lambda     | Regularisation for ridge/lasso/bayesian_ridge. Default 1.0.   |
| rank       | SVD rank for pca_impute/softimpute. Default 3L.   |
| seed       | Random seed. Default 42L.   |
| verbose    | Print progress. Default TRUE.   |

**Value**

An object of class `lmi_result`.

**See Also**

[lmi\\_impute](#), [lmi\\_methods](#)

**Examples**

```
df <- data.frame(
  age      = c(25, NA, 35, 40, NA),
  income   = c(50000, 60000, NA, 80000, 55000),
  edu      = c("BSc", NA, "MSc", "BSc", "PhD"),
  stringsAsFactors = FALSE
)
lmi_impute_offline(df)
lmi_impute_offline(df, method = "random_forest")
lmi_impute_offline(df, method = "softimpute")
```

---

`lmi_methods`*List all available offline imputation methods*

---

**Description**

Prints all 20 methods grouped by category with usage guidance.

**Usage**

```
lmi_methods()
```

**Value**

Invisibly returns a data.frame of method metadata.

**Examples**

```
lmi_methods()
```

---

`lmi_models`*List recommended models for each supported LLM provider*

---

**Description**

Prints recommended model identifiers for every provider supported by `lmi_impute`, including free-tier options. Use `lmi_set_model` to activate a model after choosing a provider with `lmi_set_api_key`.

**Usage**

```
lmi_models()
```

**Value**

Invisibly returns a character vector of all model identifiers across every provider.

**See Also**

[lmi\\_set\\_model](#), [lmi\\_providers](#), [lmi\\_impute](#)

**Examples**

```
lmi_models()
```

---

|                            |   |
|----------------------------|---|
| <code>lmi_providers</code> | <i>List all supported LLM providers</i> |
|----------------------------|---|

---

**Description**

Prints every LLM provider supported by [lmi\\_impute](#), grouped into cloud APIs, local servers, and custom endpoints, with free-tier indicators and required environment variables.

**Usage**

```
lmi_providers()
```

**Value**

Invisibly returns a character vector of provider names.

**See Also**

[lmi\\_set\\_api\\_key](#), [lmi\\_models](#)

**Examples**

```
lmi_providers()
```

---

|                              |   |
|------------------------------|---|
| <code>lmi_set_api_key</code> | <i>Configure the API key and LLM provider for llmimpute</i> |
|------------------------------|---|

---

**Description**

Sets the API key and LLM provider used by LLM-mode imputation. Supports cloud APIs (Anthropic 'Claude', OpenAI, Google Gemini, Groq, Mistral, Cohere, OpenRouter, Together AI, Fireworks AI, DeepSeek, Perplexity, xAI Grok, AI21 Labs, Cerebras) and local servers (Ollama, LM Studio, Jan, llama.cpp, KoboldCpp, Text Generation WebUI). Any other OpenAI-compatible endpoint can be used via `provider = "custom"` with a `base_url`.

If no API key is configured, [lmi\\_impute](#) automatically falls back to the offline statistical engine. The package is fully functional without any API key.

**Usage**

```
lmi_set_api_key(
    api_key = NULL,
    provider = "anthropic",
    base_url = NULL,
    .session = TRUE
)
```

**Arguments**

|          |  |
|----------|--|
| api_key  | Character string. Your API key. Not required for local providers (Ollama, LM Studio, Jan, llama.cpp, KoboldCpp, Text Generation WebUI). If NULL, the function reads from the provider's environment variable.  |
| provider | Character string. Provider name. One of: "anthropic", "openai", "gemini", "groq", "mistral", "cohere", "openrouter", "together", "fireworks", "deepseek", "perplexity", "xai", "ai21", "cerebras", "ollama", "lmstudio", "jan", "llamacpp", "koboldcpp", "textgenwebui", or "custom". Default "anthropic". |
| base_url | Character string. Required only for "custom". For local providers you can override the default port here too, e.g. base_url = "http://localhost:9000/v1/chat/completions".   |
| .session | Logical. Store for the current session only (default TRUE).  |

**Value**

Invisibly returns the API key string (or "" for keyless local providers).

**See Also**

[lmi\\_providers](#), [lmi\\_set\\_model](#), [lmi\\_impute](#)

**Examples**

```
## Not run:
## Cloud APIs
lmi_set_api_key("sk-ant-...", provider = "anthropic")
lmi_set_api_key("sk-...", provider = "openai")
lmi_set_api_key("AIza...", provider = "gemini") # free tier
lmi_set_api_key("gsk...", provider = "groq") # free tier
lmi_set_api_key("sk-or-...", provider = "openrouter") # free models
lmi_set_api_key("...", provider = "deepseek")
lmi_set_api_key("...", provider = "cerebras") # free tier

## Local servers (no key needed)
lmi_set_api_key(provider = "ollama")
lmi_set_api_key(provider = "lmstudio")
lmi_set_api_key(provider = "jan")
lmi_set_api_key(provider = "llamacpp")
lmi_set_api_key(provider = "koboldcpp")
lmi_set_api_key(provider = "textgenwebui")
```

```
## Any OpenAI-compatible endpoint
lmi_set_api_key("mykey", provider = "custom",
               base_url = "http://my-server:8000/v1/chat/completions")

## Override local port
lmi_set_api_key(provider = "ollama",
               base_url = "http://localhost:11435/api/chat")

## End(Not run)
```

---

|               |  |
|---------------|--|
| lmi_set_model | <i>Set the LLM model used for imputation</i> |
|---------------|--|

---

## Description

Sets or retrieves the model identifier used by `lmi_impute` in LLM mode. The default model is determined by the active provider. Use `lmi_models` to see recommended models per provider.

## Usage

```
lmi_set_model(model = NULL)
```

```
lmi_get_model()
```

## Arguments

|       |   |
|-------|---|
| model | Character string. A valid model identifier for the active provider. If NULL, returns the current model without changing it. |
|-------|---|

## Value

Invisibly returns the active model string.

## See Also

[lmi\\_set\\_api\\_key](#), [lmi\\_models](#)

## Examples

```
lmi_get_model()

## Not run:
lmi_set_model("gpt-4o-mini")
lmi_set_model("gemini-1.5-flash")
lmi_set_model("llama-3.3-70b-versatile")
lmi_set_model("deepseek-chat")
lmi_set_model("llama3.2")      # Ollama local
```

```
## End(Not run)
```

---

```
print.lmi_result      Print an lmi_result object
```

---

### Description

Displays a formatted summary of an imputation result in the console, including overall statistics, per-column imputation counts, and the first `n` imputed values with their confidence scores and reasoning.

### Usage

```
## S3 method for class 'lmi_result'  
print(x, n = 10L, ...)
```

### Arguments

|                  |   |
|------------------|---|
| <code>x</code>   | An object of class <code>lmi_result</code> returned by <code>lmi_impute</code> or <code>lmi_impute_offline</code> . |
| <code>n</code>   | Integer. Number of individual imputation rows to display. Default 10L.  |
| <code>...</code> | Currently unused. Included for S3 compatibility.  |

### Value

Invisibly returns `x`.

### Examples

```
df <- data.frame(  
  age    = c(25L, NA, 35L),  
  income = c(50000, 60000, NA),  
  stringsAsFactors = FALSE  
)  
result <- lmi_impute_offline(df, verbose = FALSE)  
print(result)
```

---

summary.lmi\_result      *Summarise an lmi\_result object*

---

**Description**

Returns a data.frame summarising imputation counts and confidence statistics per column, suitable for further analysis or reporting.

**Usage**

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

**Arguments**

object            An object of class lmi\_result.  
...                Currently unused. Included for S3 compatibility.

**Value**

A data.frame with columns column, n\_imputed, mean\_confidence, min\_confidence, max\_confidence. Returns NULL invisibly when no imputations were performed.

**Examples**

```
df <- data.frame(  
  age = c(25L, NA, 35L, 40L),  
  income = c(50000, 60000, NA, 80000),  
  stringsAsFactors = FALSE  
)  
result <- lmi_impute_offline(df, verbose = FALSE)  
summary(result)
```

# Index

`as.data.frame.lmi_result`, [2](#)

`lmi_diagnose`, [3](#), [7](#)  
`lmi_export`, [4](#), [7](#)  
`lmi_get_model (lmi_set_model)`, [12](#)  
`lmi_impute`, [3](#), [4](#), [5](#), [8–13](#)  
`lmi_impute_offline`, [3–6](#), [7](#), [7](#), [13](#)  
`lmi_methods`, [8](#), [9](#)  
`lmi_models`, [9](#), [10](#), [12](#)  
`lmi_providers`, [10](#), [10](#), [11](#)  
`lmi_set_api_key`, [9](#), [10](#), [10](#), [12](#)  
`lmi_set_model`, [9–11](#), [12](#)

`print.lmi_result`, [7](#), [13](#)

`summary.lmi_result`, [14](#)