Package: DataDNA (via r-universe)

November 3, 2026

Title Data Frame Fingerprints and Lineage Figures

Version 0.1.0

Description Profiles R data frames as compact data fingerprints using schema, shape, missingness, distribution, category, uniqueness, time, and role signals. It compares versions, identifies close relatives in a library of historical data sets, and renders portable HTML cards plus static PNG/PDF lineage figures for reports.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports htmltools

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

Date/Publication 2026-05-04 10:13:22 UTC

RemoteUrl https://github.com/tonyisfool/datadna

RemoteRef HEAD

RemoteSha 590aa03c4ecf67c3213cb41b95a2ea335101e104

Contents

customers_new 2
customers_old 2
data_dna 3
dna_card 3
dna_compare 4
dna_diff 4
dna_example_customers 5
dna_match 5

dna_match_card 6
dna_match_plot 7
dna_species 7

Index **9**

`customers_new` *New customer table for DataDNA examples*

Description

A modified version of `customers_old` with distribution, category, missingness, and schema changes.

Format

A data frame with 180 rows and 9 columns.

Source

Synthetic data generated for package examples.

`customers_old` *Old customer table for DataDNA examples*

Description

A small synthetic customer table used to demonstrate data DNA profiling.

Format

A data frame with 180 rows and 8 columns.

Source

Synthetic data generated for package examples.

data_dna *Create a data DNA profile*

Description

Profiles an R data frame into a compact identity object that records schema, shape, missingness, distributions, categories, uniqueness, time signals, and stable fingerprints.

Usage

```
data_dna(df, name = NULL, sample_size = 10000L)
```

Arguments

df	A data frame.
name	Optional data set name shown on cards and print output.
sample_size	Maximum number of rows used for profiling.

Value

A data_dna object.

Examples

```
demo <- dna_example_customers()
dna <- data_dna(demo$customers_new, name = "customers_new")
dna
```

dna_card *Render a laboratory-style data DNA card*

Description

Render a laboratory-style data DNA card.

Usage

```
dna_card(x, file = NULL, open = FALSE)
```

Arguments

x	A data frame or data_dna object.
file	Optional HTML file path. If supplied, the card is saved there.
open	Logical. Open the saved file in a browser when file is supplied.

Value

An `htmltools` browsable object, invisibly when saved to `file`.

Examples

```
demo <- dna_example_customers()
card <- dna_card(demo$customers_new)
```

`dna_compare` *Compare two data DNA profiles*

Description

Compare two data DNA profiles.

Usage

```
dna_compare(x, y)
```

Arguments

`x`	A data frame or `data_dna` object.
`y`	A data frame or `data_dna` object.

Value

A `dna_comparison` object.

Examples

```
demo <- dna_example_customers()
dna_compare(demo$customers_old, demo$customers_new)
```

`dna_diff` *Explain mutations between two data DNA profiles*

Description

Explain mutations between two data DNA profiles.

Usage

```
dna_diff(x, y)
```

Arguments

x	A data frame or data_dna object.
y	A data frame or data_dna object.

Value

A dna_diff object containing a mutation table.

Examples

```
demo <- dna_example_customers()
dna_diff(demo$customers_old, demo$customers_new)
```

---

dna_example_customers *Example customer tables*

Description

Creates two small customer data frames designed to demonstrate DataDNA cards, comparison, and mutation reports.

Usage

```
dna_example_customers()
```

Value

A list with customers_old and customers_new data frames.

Examples

```
demo <- dna_example_customers()
str(demo$customers_old)
```

---

dna_match *Match a data set against a DNA library*

Description

Finds the closest relatives of a query data set by comparing its data DNA against a named library of data frames or data_dna objects.

Usage

```
dna_match(x, library, top_n = 5L, sample_size = 10000L)
```

Arguments

x	A data frame or data_dna object to match.
library	A list of data frames or data_dna objects.
top_n	Maximum number of matches to return.
sample_size	Maximum number of rows used when profiling raw data frames.

Value

A dna_match object.

Examples

```
demo <- dna_example_customers()
lib <- list(old = data_dna(demo$customers_old), new = data_dna(demo$customers_new))
dna_match(demo$customers_new, lib)
```

---

dna_match_card *Render a DataDNA lineage match card*

Description

Creates a static HTML/SVG lineage diagram for a dna_match object.

Usage

```
dna_match_card(match, file = NULL, open = FALSE)
```

Arguments

match	A dna_match object.
file	Optional HTML file path. If supplied, the card is saved there.
open	Logical. Open the saved file in a browser when file is supplied.

Value

An htmltools browsable object, invisibly when saved to file.

Examples

```
demo <- dna_example_customers()
lib <- list(old = data_dna(demo$customers_old), new = data_dna(demo$customers_new))
match <- dna_match(demo$customers_new, lib)
dna_match_card(match)
```

dna_match_plot *Draw a paper-style lineage figure*

Description

Creates a print-friendly, paper-style lineage figure for a dna_match object using base R grid graphics. The figure can be drawn on the current graphics device or saved directly to PNG or PDF.

Usage

```
dna_match_plot(match, file = NULL, width = 11, height = 7, dpi = 144)
```

Arguments

match	A dna_match object.
file	Optional output path. Supported extensions are .png and .pdf.
width	Plot width in inches.
height	Plot height in inches.
dpi	Resolution used for PNG output.

Value

The input dna_match object, invisibly.

Examples

```
demo <- dna_example_customers()
lib <- list(old = data_dna(demo$customers_old), new = data_dna(demo$customers_new))
match <- dna_match(demo$customers_new, lib)
dna_match_plot(match)
```

dna_species *Guess the species of a data frame*

Description

Guess the species of a data frame.

Usage

```
dna_species(df)
```

Arguments

df	A data frame.

Value

A character label such as `customer_table`, `event_stream`, or `wide_feature_matrix`.

Examples

```
dna_species(dna_example_customers()$customers_new)
```

Index

∗ **datasets**
customers_new, 2
customers_old, 2

customers_new, 2
customers_old, 2

data_dna, 3
dna_card, 3
dna_compare, 4
dna_diff, 4
dna_example_customers, 5
dna_match, 5
dna_match_card, 6
dna_match_plot, 7
dna_species, 7