Package: modelname (via r-universe)

October 16, 2026

Title Parse AI Model Names into Vendor, Family and Version

Version 0.1.0

Description Splits the identifiers and display names that AI labs give their models, such as 'claude-opus-4-1-20250805', 'gpt-4.1-mini' or 'Gemini 3.1 Pro Preview', into vendor, family, version, parameter size, release date and tags. Old and new naming schemes from 'Anthropic', 'OpenAI', 'Google' and 'OpenRouter' land in the same family, so models can be compared and the newest one in a family picked from a list. A port of the 'Go' module 'modelname' <https://pkg.go.dev/github.com/BoNour/modelname>.

License MIT + file LICENSE

URL https://whizi.io/models/, https://github.com/BoNour/modelname-r

BugReports https://github.com/BoNour/modelname-r/issues

Depends R (>= 3.5.0)

Suggests tinytest

Encoding UTF-8

NeedsCompilation no

Author Nour Ayman [aut, cre, cph]

Maintainer Nour Ayman <nourayman850@gmail.com>

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

Date/Publication 2026-10-16 13:20:15 UTC

RemoteUrl https://github.com/cran/modelname

RemoteRef HEAD

RemoteSha 93ccee28e0d4f246f2effe6fcd536b7d0584ecd3

Contents

modelname-package 2
compare_models 2

latest_model 3
parse_model 4
same_family 5

Index **6**

`modelname-package` *Parse AI Model Names into Vendor, Family and Version*

Description

Parses AI model identifiers and display names into parts you can compare. It is a port of the 'Go' module 'modelname', with the same rules and the same test cases.

Details

`parse_model` does the parsing. `same_family`, `compare_models` and `latest_model` build on it.

See Also

https://whizi.io/models/ lists the real model names most of the test cases come from.

Examples

```
parse_model("Qwen3.6 35B A3B")
```

`compare_models` *Order Two Model Names*

Description

Orders two models from the same family, oldest first. The version wins, then the full date, then the short date code. Version parts are compared as numbers, so `"3.10"` is newer than `"3.9"` and `"k3"` is newer than `"k2.7"`.

Usage

```
compare_models(a, b)
```

Arguments

a, b	Character vectors of model identifiers or display names. The shorter one is recycled.

Details

Comparing across families works but does not mean much: `"gpt-5"` against `"claude-sonnet-4-5"` just compares 5 with 4.5. Check `same_family` first if that matters.

Value

An integer vector: `-1` where `a` is older, `1` where `a` is newer and `0` where the two cannot be told apart. `NA` where either input is `NA`.

See Also

`parse_model`, `latest_model`

Examples

```
compare_models("gemini-3.9-flash", "gemini-3.10-flash")
compare_models("claude-opus-4-1-20250901", "claude-opus-4-1-20250805")
compare_models("gpt-5", "GPT-5")
```

---

latest_model — *Pick the Newest Model in a Family*

Description

Returns the newest identifier in `ids` whose family matches. Handy for picking a default out of a provider's model list.

Usage

```
latest_model(ids, family)
```

Arguments

`ids`	A character vector of model identifiers or display names.
`family`	A single string, the family to look for, as returned in the `family` column of `parse_model`.

Value

A single string, the element of `ids` that is newest according to `compare_models`, or `NA` if no element is in `family`. Ties go to the first one in `ids`.

See Also

`parse_model`, `compare_models`

Examples

```
ids <- c("claude-sonnet-4-5", "claude-3-7-sonnet-20250219",
         "claude-sonnet-4-6", "gpt-5")
latest_model(ids, "claude-sonnet")
latest_model(ids, "llama")
```

`parse_model` *Parse Model Names*

Description

Splits AI model identifiers and display names into vendor, family, version, parameter size, date, date code and tags. API identifiers such as `"gpt-4.1-mini"`, 'OpenRouter' style paths such as `"meta-llama/llama-3.3-70b-instruct:free"` and display names such as `"Gemini 3.1 Pro Preview"` are all read the same way.

Usage

```
parse_model(x)
```

Arguments

`x`	A character vector of model identifiers or display names. Factors are converted to character.

Details

Parsing is heuristic, not a registry lookup. It never fails: anything it does not recognise ends up in `family`, and fields that are not in the name are left empty. Old and new naming schemes land in the same family, so `"claude-3-5-sonnet-20241022"` and `"Claude Sonnet 4.5"` both have family `"claude-sonnet"`.

The vendor comes from the prefix before the last `/` if there is one, otherwise from the first word of the family. Unknown first words give an empty vendor.

Value

A data frame with one row per element of `x` and these columns:

id The input, unchanged.

vendor Who makes the model, such as `"anthropic"` or `"openai"`, or `""` if unknown.

family Name words joined with `-`, such as `"gemini-flash-lite"`.

version Version string, such as `"4.5"`, `"4o"` or `"k2.7"`.

size Parameter count if given, such as `"70b"` or `"35b-a3b"`.

date A `Date`, only set for full dates like `20241022`, otherwise `NA`.

snapshot Short date codes like `"0731"` or `"2512"`.

tags A list column of character vectors, such as `"preview"`, `"latest"`, `"thinking"` or the `":free"` suffix.

Missing character fields are `""`. An `NA` input gives a row of `NA` values.

See Also

`compare_models`, `same_family`, `latest_model`

Examples

```
parse_model("claude-opus-4-1-20250805")

m <- parse_model(c("gpt-4.1-mini", "meta-llama/llama-3.3-70b-instruct:free",
                   "Gemini 3.1 Pro Preview"))
m[, c("vendor", "family", "version", "size")]
m$tags
```

`same_family` *Check Whether Two Model Names Share a Family*

Description

Parses both arguments with `parse_model` and compares their `family` columns.

Usage

```
same_family(a, b)
```

Arguments

a, b	Character vectors of model identifiers or display names. The shorter one is recycled.

Value

A logical vector, `TRUE` where the two names parse to the same family. `NA` where either input is `NA`.

See Also

`parse_model`, `compare_models`

Examples

```
same_family("claude-3-5-sonnet-20241022", "Claude Sonnet 4.5")
same_family("gpt-4.1-mini", c("GPT-5.4 Mini", "gpt-4.1"))
```

Index

compare_models, 2, *2–5*

latest_model, *2*, 3, *3, 4*

modelname *(*modelname-package*)*, 2
modelname-package, 2

parse_model, *2, 3*, 4, *5*

same_family, *2*, *4*, 5