

# Package ‘ChineseNames’

May 7, 2026

**Title** Chinese Name Database 1930-2008

**Version** 2025.8

**Date** 2025-08-15

**Maintainer** Han Wu Shuang Bao <baohws@foxmail.com>

**Description** A database of Chinese surnames and given names (1930-2008).

This database contains nationwide frequency statistics of 1,806 Chinese surnames and 2,614 Chinese characters used in given names, covering about 1.2 billion Han Chinese population (96.8 percent of the Han Chinese household-registered population born from 1930 to 2008 and still alive in 2008).

This package also contains a function for computing multiple indices of Chinese surnames and given names for social science research (e.g., name uniqueness, name gender, name valence, and name warmth/competence). Details are provided at

<<https://psychbruce.github.io/ChineseNames/>>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**URL** <https://psychbruce.github.io/ChineseNames/>

**BugReports** <https://github.com/psychbruce/ChineseNames/issues>

**Depends** R (>= 4.0.0)

**Imports** bruceR, data.table

**Suggests** babynames, car, dplyr, glue

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Han Wu Shuang Bao [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-3043-710X>>)

**Repository** CRAN

**Date/Publication** 2025-08-19 16:00:07 UTC

## Contents

compute_name_index . . . . .	2
familyname . . . . .	4
givenname . . . . .	5
population . . . . .	6
top1000name.prov . . . . .	6
top100name.year . . . . .	6
top50char.year . . . . .	7
<b>Index</b>	<b>8</b>

---

compute_name_index	<i>Compute multiple indices of surnames and given names.</i>
--------------------	--

---

## Description

Compute all available name features (indices) based on `familyname` and `givenname`. You can either input a data frame with a variable of Chinese full names (and a variable of birth years, if necessary) or just input a vector of full names (and a vector of birth years, if necessary).

- Usage 1: Input a single value or a vector of name (and birth, if necessary).
- Usage 2: Input a data frame of data and the variable name of `var.fullname` (or `var.surname` and/or `var.givenname`) (and `var.birthyear`, if necessary).

*Caution:* Name-character uniqueness (NU) for birth year  $\geq 2010$  is estimated by forecasting and thereby may not be accurate.

## Usage

```
compute_name_index(
  data = NULL,
  var.fullname = NULL,
  var.surname = NULL,
  var.givenname = NULL,
  var.birthyear = NULL,
  name = NA,
  birth = NA,
  index = c("NLen", "SNU", "SNI", "NU", "CCU", "NG", "NV", "NW", "NC"),
  NU.approx = TRUE,
  digits = 4,
  return.namechar = TRUE,
  return.all = FALSE
)
```

**Arguments**

data	Data frame.
var.fullname	Variable name of Chinese full names (e.g., "name").
var.surname	Variable name of Chinese surnames (e.g., "surname").
var.givename	Variable name of Chinese given names (e.g., "givename").
var.birthyear	Variable name of birth year (e.g., "birth").
name	If no data, you can just input a vector of full name(s).
birth	If no data, you can just input a vector of birth year(s).
index	Which indices to compute? By default, it computes all available name indices: <ul style="list-style-type: none"> <li>• NLen: full-name length (2~4).</li> <li>• SNU: surname uniqueness (1~6).</li> <li>• SNI: surname initial (1~26).</li> <li>• NU: name-character uniqueness (1~6).</li> <li>• CCU: character-corpus uniqueness (1~6).</li> <li>• NG: name gender (-1~1).</li> <li>• NV: name valence (1~5).</li> <li>• NW: name warmth (1~5).</li> <li>• NC: name competence (1~5).</li> </ul>
NU.approx	Whether to <i>approximately</i> compute name-character uniqueness (NU) using <i>the nearest two birth cohorts with relative weights</i> (which would be more precise than just using a single birth cohort). Defaults to TRUE.
digits	Number of decimal places. Defaults to 4.
return.namechar	Whether to return separate name characters. Defaults to TRUE.
return.all	Whether to return all temporary variables in the computation of the final variables. Defaults to FALSE.

**Details**

<https://psychbruce.github.io/ChineseNames/>

**Value**

A new data frame (class `data.table`) with name indices appended. Full names are split into `name0` (surnames, with compound surnames automatically detected), `name1`, `name2`, and `name3` (given-name characters).

**Examples**

```
## Prepare ##
sn = familyname$surname[1:12]
gn = c(top100name.year$name.all.1960[1:6],
       top100name.year$name.all.2000[1:6],
```

```

      top100name.year$name.all.1960[95:100],
      top100name.year$name.all.2000[95:100])
demodata = data.frame(name=paste0(sn, gn),
                      birth=c(1960:1965, 2000:2005,
                              1960:1965, 2000:2005))

demodata

## Compute ##
newdata = compute_name_index(demodata,
                             var.fullname="name",
                             var.birthyear="birth")

newdata

```

---

familyname	<i>1,806 Chinese surnames and nationwide frequency.</i>
------------	---

---

### Description

1,806 Chinese surnames and nationwide frequency.

### Usage

```
data(familyname)
```

### Format

A data frame with 7 variables:

surname surname (in Chinese)  
 compound 0 = single surname, 1 = compound surname  
 initial initial letter (a-z)  
 initial.rank initial order (1-26)  
 n.1930\_2008 total counts in the database  
 ppm.1930\_2008 proportion in population (ppm = parts per million)  
 surname.uniqueness surname uniqueness

### Details

<https://psychbruce.github.io/ChineseNames/>

---

givenname	<i>2,614 Chinese characters used in given names and nationwide frequency.</i>
-----------	---

---

**Description**

2,614 Chinese characters used in given names and nationwide frequency.

**Usage**

```
data(givenname)
```

**Format**

A data frame with 25 variables:

character character used in given names (in Chinese)

pinyin pinyin (pronunciation)

bihua number of strokes in a character

n.male total counts in male

n.female total counts in female

name.gender difference in proportions of a character used by male vs. female

n.1930\_1959, n.1960\_1969, n.1970\_1979, n.1980\_1989, n.1990\_1999, n.2000\_2008 total counts in a birth cohort

ppm.1930\_1959, ppm.1960\_1969, ppm.1970\_1979, ppm.1980\_1989, ppm.1990\_1999, ppm.2000\_2008 proportion (parts per million) in a birth cohort

name.ppm average ppm (parts per million) across all cohorts

name.uniqueness name-character uniqueness (in naming practices)

corpus.ppm proportion (parts per million) in contemporary Chinese corpus

corpus.uniqueness character-corpus uniqueness (in contemporary Chinese corpus)

name.valence name valence (positivity of character meaning) (based on subjective ratings from 16 raters, ICC = 0.921)

name.warmth name warmth/morality (based on subjective ratings from 10 raters, ICC = 0.774)

name.competence name competence/assertiveness (based on subjective ratings from 10 raters, ICC = 0.712)

**Details**

<https://psychbruce.github.io/ChineseNames/>

population                      *Population statistics for the Chinese name database.*

---

**Description**

Population statistics for the Chinese name database.

**Usage**

```
data(population)
```

**Details**

<https://psychbruce.github.io/ChineseNames/>

---

top1000name.prov                *Top 1,000 given names in 31 Chinese mainland provinces.*

---

**Description**

Top 1,000 given names in 31 Chinese mainland provinces.

**Usage**

```
data(top1000name.prov)
```

**Details**

<https://psychbruce.github.io/ChineseNames/>

---

top100name.year                *Top 100 given names in 6 birth cohorts.*

---

**Description**

Top 100 given names in 6 birth cohorts.

**Usage**

```
data(top100name.year)
```

**Details**

<https://psychbruce.github.io/ChineseNames/>

---

top50char.year	<i>Top 50 given-name characters in 6 birth cohorts.</i>
----------------	---

---

### **Description**

Top 50 given-name characters in 6 birth cohorts.

### **Usage**

```
data(top50char.year)
```

### **Details**

<https://psychbruce.github.io/ChineseNames/>

# Index

`compute_name_index`, [2](#)

`familyname`, [2, 4](#)

`givenname`, [2, 5](#)

`population`, [6](#)

`top1000name.prov`, [6](#)

`top100name.year`, [6](#)

`top50char.year`, [7](#)