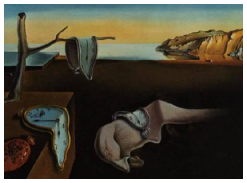

Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

regioncode: Convert Region Names and Division Codes of China Over Years

Yue Hu
Tsinghua University

Wenquan Wu
Tsinghua University,

Abstract

The Chinese government gives unique geocodes for each county, city (prefecture), and provincial-level administrative unit. `regioncode` provides a convenient way to convert Chinese administrative division codes, official names, sociopolitical and linguistic areas, abbreviations, and so on between each other.

Keywords: regioncode, geocodes, administrative division codes, linguistic zone, Pinyin.

1. Why regioncode?

The Chinese government gives unique geocodes for each county, city (prefecture), and provincial-level administrative unit. These "administrative division codes" are consistently adjusted and updated to matched national and regional plans of development. The adjustments however may disturb researchers when they conduct studies over time or merge geo-based data from different years. Especially, when researchers render statistical data on a Chinese map, different geocodes between map data and statistical data can cause mess-up outputs.

This package aims to conquer such difficulties by a one-step solution. In the current version, `regioncode` enables seamlessly converting formal names, common-used names, language zone, and division codes of Chinese provinces and prefectures between each other and across thirty-four years from 1986 to 2019.

2. Installation

To install:

- the latest released version: `install.packages("regioncode")`.
- the latest developing version: `remotes::install_github("sammo3182/regioncode")`.

3. Basic Usage

We uses a randomly drawn sample of Yuhua Wang's China's Corruption Investigations Dataset to illustrate how the package works. In the data, the division codes were recorded with the 2019 version, and we added prefectural abbreviations for the sake of illustration.

In `regioncode` package, we named administrative division codes as `code`, regions' formal names as `name`, and their commonly used abbreviation as `sname`. The current version enables mutual conversion between any pair of them. To do so, users just need to pass a character vector of names or a numeric vector of geocodes into the function. In the current version, the function can produce three types of output at both the prefectural and provincial levels: codes (`code`), names (`name`) and area (`area`, such as 华北, 东北, 华南, etc.). One just needs to specify which type of the output they want in the argument `convert_to` and corresponding years of the input and output. For example, the following codes convert the 2019 geocodes in the `corruption` data to their 1989 version:

```
library(regioncode)
```

```
Warning: package 'regioncode' was built under R version 4.3.1
```

```
data("corruption")

# Original 2019 version
corruption$prefecture_id
```

```
 [1] 370100 321200 310117 420500 451300 431200 350300 511500 469021 420600
```

```
# 1999 version
regioncode(data_input = corruption$prefecture_id,
           convert_to = "code", # default set
           year_from = 2019,
           year_to = 1989)
```

```
Joining with `by = join_by(`2019_code`)`
```

```
 [1] 370100 329001 310227 420500 422700 452200 433000 350300 512500 460025
[11] 420600
```

Note that if a region was initially geocoded in e.g., 1989 and included in a new region, in 2019, the new region geocode will be used hereafter. If a big place was broken into several

regions, the later-year codes will be aligned with the first region according to the ascendant order of the regions' numeric geocodes.

By altering the output format to `name`, one can easily convert codes or region names of a given year to region names in another year. `regioncode` automatically detects the input format, so users need to specify the *output* format only (together with the input and output years) to gain what they want. In the following example, we convert the geocode variable in the `corruption` dataset to region names and the name variable to codes and names in another year.

```
# The original name
corruption$prefecture
```

```
 [1] "济南市" "泰州市" "松江区" "宜昌市" "来宾市" "怀化市" "莆田市" "宜宾市"
 [9] "定安县" "襄阳市"
```

```
# Codes to name

regioncode(data_input = corruption$prefecture_id,
           convert_to = "name",
           year_from = 2019,
           year_to = 1989)
```

```
Joining with `by = join_by(`2019_code`)`
```

```
 [1] "济南市"   "泰州市"   "松江县"   "宜昌市"   "宜昌地区" "柳州地区"
 [7] "怀化地区" "莆田市"   "宜宾地区" "定安县"   "襄樊市"
```

```
# Name to codes of the same year
regioncode(data_input = corruption$prefecture,
           convert_to = "code",
           year_from = 2019,
           year_to = 2019)
```

```
Joining with `by = join_by(`2019_name`)`
```

```
 [1] 370100 321200 310117 420500 451300 431200 350300 511500 469021 420600
```

```
# Name to name of a different year

regioncode(data_input = corruption$prefecture,
           convert_to = "name",
           year_from = 2019,
           year_to = 1989)
```

```
Joining with `by = join_by(`2019_name`)`
```

```
 [1] "济南市"   "泰州市"   "松江县"   "宜昌市"   "宜昌地区" "柳州地区"
 [7] "怀化地区" "莆田市"   "宜宾地区" "定安县"   "襄樊市"
```

4. Advanced Applications

To further help uses with "messier" data and diverse demands, `regioncode` provides five special conversions: conversion from data with incomplete data, specification of municipalities, conversion sociopolitical areas and linguistic areas, and pinyin output. The current version also allows conversions at the provincial level.

4.1. Incomplete naming prefectures.

More than often, data codes may omit the administrative level when recording geo-information, e.g., using "北京" instead of "北京市." To accomplish conversions of such data, one needs to specify the `incomplete_name` argument. If the input data is incomplete, users should set the argument as "from"; if they want the output name (when `convert_to = "name"`) to be without "city" or "prefecture," they can set the argument to "to" (see the example below); and if users want to gain incomplete names for both input and output names, `incomplete_name = "both"`. All the above conversions can be over years.

```
# Full, official names
corruption$prefecture
```

```
 [1] "济南市" "泰州市" "松江区" "宜昌市" "来宾市" "怀化市" "莆田市" "宜宾市"
 [9] "定安县" "襄阳市"
```

```
regioncode(data_input = corruption$prefecture,
           convert_to = "name",
           year_from = 2019,
           year_to = 1989,
           incomplete_name = "to")
```

```
Joining with `by = join_by(`2019_name`)`
```

```
 [1] "济南" "泰州" "松江" "宜昌" "柳州" "来宾" "怀化" "莆田" "宜宾" "定安"
[11] "襄樊" "襄阳"
```

4.2. Municipalities

Municipalities ("直辖市") are geographically cities but administratively provincial. The districts within these municipalities are thus prefectural. Different analyses treat these districts

differently: some parallel the districts aligned with other prefectures, while the others treat the entire municipality as one prefecture. To deal with the latter situation, `regioncode` sets an argument `zhixiashi`. When the argument is set `TRUE`, the municipalities are treated as whole prefectures, and their provincial codes are used as the geocodes.

4.3. Sociopolitical and Linguistic Areas

Due to social, political, and martial reasons, Chinese regions are divided into seven regions:

region	provincial-level administrative unit
华北	北京市, 天津市, 山西省, 河北省, 内蒙古自治区
东北	黑龙江省, 吉林省, 辽宁省
华东	上海市, 江苏省, 浙江省, 安徽省, 福建省, 台湾省, 江西省, 山东省
华中	河南省, 湖北省, 湖南省
华南	广东省, 海南省, 广西壮族自治区, 香港特别行政区, 澳门特别行政区
西南	重庆市, 四川省, 贵州省, 云南省, 西藏自治区
西北	陕西省, 甘肃省, 青海省, 宁夏回族自治区, 新疆维吾尔自治区

`regioncode` also offers a method "area" to convert codes and names of the region into such areas.

```
regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1989,
           convert_to="area")
```

```
Joining with `by = join_by(`2019_name`)`
```

```
 [1] "华东" "华东" "华东" "华中" "华南" "华中" "华东" "西南" "华南" "华中"
```

China is a multilingual country with a variety of dialects. These dialects may be used by several prefectures in a province or province. Prefectures from different provinces may also share the same dialect.

`regioncode` allows users to gain linguistic zones the prefectures belong as an output. Users can gain two levels of linguistic zones, dialect groups and dialect sub-groups by setting the argument `to_pinyin` to `dia_group` or `dia_sub_group`. Note that, the linguistic distribution in China is too complex for precisely gauging at the prefectural level. The linguistic zone output from `regioncode` is thus at most for reference rather than rigorous linguistic research.

```
regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1989,
           to_dialect = "dia_group")
```

```
Joining with `by = join_by(`2019_name`)`

 [1] "冀鲁官话" "江淮官话" "吴语"     "西南官话" "西南官话" "湘语"
 [7] "莆仙区"   "西南官话" "琼文区"   "西南官话"

regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1989,
           to_dialect = "dia_sub_group")

Joining with `by = join_by(`2019_name`)`

 [1] "沧惠片-1"  "石济片-8"  "泰如片-1"  "太湖片-1"  "成渝片-3"  "成渝片-9"
 [7] "桂柳片-10" "岑江片-2"  "吉溆片-3"  "娄邵片-1"  "黔北片-3"  "长益片-3"
[13] "莆仙区-4"  "灌赤片-10" "府城片-1"  "鄂北片-10"
```

4.4. Pinyin

Pinyin is a Chinese phonetic romanization. Some data stores the region names with pinyin instead of Chinese characters. The default name output of `regioncode` uses Chinese characters, but one can gain pinyin output by setting the argument `to_pinyin = TRUE`. The effect can be applied to either official name, incomplete name, or sociopolitical area outputs.

```
regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1989,
           convert_to="name",
           to_pinyin=TRUE
           )

Joining with `by = join_by(`2019_name`)`

          济南市           泰州市           松江县           宜昌市
    "ji_nan_shi"  "tai_zhou_shi" "song_jiang_xian"  "yi_chang_shi"
        宜昌地区         柳州地区         怀化地区         莆田市
"yi_chang_di_qu" "liu_zhou_di_qu" "huai_hua_di_qu"   "pu_tian_shi"
        宜宾地区           定安县           襄樊市
  "yi_bin_di_qu"  "ding_an_xian" "xiang_fan_shi"

regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1989,
           convert_to="name",
           incomplete_name = "to",
           to_pinyin=TRUE
           )
```

```
Joining with `by = join_by(`2019_name`)`
```

```
      济南       泰州       松江       宜昌       柳州       来宾
 "ji_nan" "tai_zhou" "song_jiang" "yi_chang" "liu_zhou" "lai_bin"
      怀化       莆田       宜宾       定安       襄樊       襄阳
"huai_hua"  "pu_tian"   "yi_bin"  "ding_an" "xiang_fan" "xiang_yang"
```

```
regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1989,
           convert_to="area",
           to_pinyin=TRUE
           )
```

```
Joining with `by = join_by(`2019_name`)`
```

```
      华东       华东       华东       华中       华南       华中
"hua_dong" "hua_dong" "hua_dong" "hua_zhong"  "hua_nan" "hua_zhong"
      华东       西南       华南       华中
"hua_dong"  "xi_nan"  "hua_nan" "hua_zhong"
```

4.5. Provinces

`regioncode` allows conversions at not only the prefectural but provincial level. By setting the argument `province = TRUE`, users can accomplish all the code, name, and area conversions at the provincial level. (Note that, at the provincial level, the linguistic conversion can be only to dialect group.) Moreover, since provinces have fixed abbreviations, `regioncode` allows names not only being, e.g., "宁夏" instead of "宁夏回族自治区" but also "宁". When the inputs are abbreviations, users can set the `convert_to` argument to `abbreTocode`, `abbreToname`, or `abbreToarea`. When they want provincial abbreviation outputs, just set `convert_to = "abbre"`.

```
regioncode(data_input = corruption$province_id,
           convert_to = "codeToabbre",
           year_from = 2019,
           year_to = 1989,
           province = TRUE)
```

```
Joining with `by = join_by(prov_code)`
```

```
 [1] "鲁" "苏" "沪" "鄂" "桂" "湘" "闽" "蜀" "琼" "鄂"
```

5. Conclusion

`regioncode` provides a convenient way to convert Chinese administrative division codes, official names, sociopolitical and linguistic areas, abbreviations, and so on between each other. This vignette offers a quick view of package features and a short tutorial for users.

The development of the package is ongoing. Future versions aim to add more administrative level choices, from province level to county level. Data are also enriching. Welcome to join us if you are also interested (see the affiliations below). Please contact us with any questions or comments. Bug reports can be conducted by Github Issues.

Also thanks ZHU Meng and LIU Xueyan for helping writing the 'Advanced Application' section of this vignette.

References

Affiliation:

Yue Hu
E-mail: yuehu@tsinghua.edu.cn

Wenquan Wu
E-mail: wuwq20@mails.tsinghua.edu.cn

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd