The Journal of Open Source Software

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

Yue Hu[1], **Yufei Sun**[1], **and Wenquan Wu**[1]

1 Department of Political Science, Tsinghua University

DOI: 10.21105/joss.0XXXX

Software

- Review
- Repository
- Archive

Editor: Editor Name

Submitted: 01 January XXXX
Published: 01 January XXXX

Summary

The Chinese government gives unique geocodes for each county, city (prefecture), and provincial-level administrative unit. The so-called "administrative division codes" were consistently adjusted to matched national and regional plans of development. Geocode adjustments disturb researchers when they merge data with different versions of geocodes or region names. Especially, when researchers render statistical data on a Chinese map, different geocodes between map data and statistical data may cause mess-up data output or visualization.

The package is developed to conquer such difficulties to match regional data across years more conveniently and correctly. Inspired by Vincent Arel-Bundock's well-known `countrycode` (Arel-Bundock et al., 2018), we created `regioncode` to achieve similar functions specifically for China studies. `regioncode` enables seamlessly converting formal names, common-used names, and division codes of Chinese prefecture regions between each other and across thirty-four years from 1986 to 2019.

In the current version, we provide three basic functions`2code`, `2name`, and `2sname` to convert formal names, common-used names, and division codes between each other. We also provide some useful features: `incompleteName`, which completes incomplete parameters; `2area`, which converts region codes and names of the region into the municipal area that they belong to; `topinyin`, which convert the names or areas into the form of pinyin; and `todialect`, which offers a function to convert the name of prefecture from any year to language(local dialect) zone.

Examples

Division codes across years

regioncoce function accept numeric and character vectors as the input division codes and region names respectively. To achieve an accurate conversion, users have to specify the year of the source data correctly in the argument year_from. Then they can set the year they want the output is. That's it. See the following example to convert the 2019-version codes to the 1999 version:

```
# original geocodes. It's 2019 version
corruption$prefecture_id

# after conversion. It's 1999 version
regioncode(data_input = corruption$prefecture_id,
           year_from = 2019,
           year_to = 1999)
```

Division codes to region name

In some cases, the original data may only have division codes or region names, but users need the other form or both formats of data. In such cases, `regioncode` offers a function to convert division codes from any year to region names in any year. Users only need to alter the converting method, for example, to "2name" in order to convert division codes to region names.

```
regioncode(data_input = corruption$prefecture_id,
           year_from = 2019,
           year_to = 1999,
           method = "2name")
```

Similarly, one can get the code from names, or in a less-often case get the names in a different year from the names from a given year. Users need to change the method argument to "2code" or "2name" to achieve these conversions.

```
corruption$prefecture

regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1999,
           method = "2code")

regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1999,
           method = "2name")
```

Advanced Usages

Completion

`regioncode` provides two advanced functions to achieve more complicated conversions. One of the occasions occurs when the data source includes only common-used short names of the cities instead of the full, official ones. `regioncode` can still accomplish the conversion in this case when the users specify the `incompleteName` to `from`. (`regioncode` can also produce short names from inputs of full or short names and division code. See the details of the help file for more information.)

```
# Full, official names
corruption$prefecture

# Incomplete names
corruption$prefecture_sname

# Converting
regioncode(data_input = corruption$prefecture_sname,
           year_from = 2019,
           year_to = 1999,
           method = "2code",
           incompleteName = "from")
```

Municipalities

Another advanced application involves the case when the municipalities directly under the central government ("zhixiashi" in Chinese Pinyin). This is common for national survey data. regioncode can fit this case with no problem as long as the user sets the argument zhixiashi as TRUE.

```
# In the sample data, the division code of municipalities were coded as NA. Fillin
code_zhixiashi <- c("110000", "120000", "310000", "400000")

corruption <- corruption %>%
  mutate(prefecture_id = ifelse(province_id %in% code_zhixiashi, province_id, pref

## Converting

regioncode(data_input = corruption$prefecture_id,
           year_from = 2019,
           year_to = 1999,
           zhixiashi = TRUE)
```

2area

regioncode also offers a method "2area" to convert codes and names of the region into the municipal area that they belong to.

```
regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1999,
           province = F,
           method="2area")
```

2pinyin

regioncode offers a parameter "topinyin" to convert the names or areas into the form of pinyin. The default of topinyin is set as FALSE, and only when the output form is character that the converting process will begin.

```
regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1999,
           province = F,
           method="2area",
           topinyin=FALSE
           )
```

2dialect

regioncode also offers a function to convert name of prefecture from any year to language zone. Users need to change the `todialect` argument to "dia_group" or "dia_sub_group" to achieve these transformations. Similarly, one can get the language zone from the province name. As long as the user sets the argument `province` as TRUE and changes the `todialect` argument to "dia_super."

```
regioncode(data_input = corruption$prefecture,
           year_from = 2019,
           year_to = 1999,
           province = F,
           todialect = "dia_group")
```

Acknowledgements

We acknowledge contributions from Meng Zhu, Yuyang Shi, Yujia Xu, and Yuxin Pan, Haiting Tian, Weihang Shao, and Yuanqian Chen.

References

Arel-Bundock, V., Enevoldsen, N., & Yetman, C. (2018). Countrycode: An r package to convert country names and country codes. *Journal of Open Source Software*, *3*(28), 848. https://doi.org/10.21105/joss.00848