Changelog

Peng Huang

2024-10-09

This is a supporting document for the data and code of **IRS-migration-data-2.0**. It details the changes made to the original version of the data published by **Hauer & Byars (2019)** on **GitHub**.

Correct FIPS coding in Year 1998

The previous dataset had some error in generating the five-digit state+county FIPS code. As an indicator of the issue, some state+county FIPS codes only had three digits rather than five, and one can detect this using the following code:

```r
irs <- read.table("./DATA-PROCESSED/county_migration_data.txt",header=TRUE)
no.state.fips <-  floor(irs$origin/1e3)==0 | floor(irs$destination/1e3)==0
table(no.state.fips)
head(irs[no.state.fips,])
```

The reason of the error comes from the fact that while most data files store FIPS codes as text, some data files store FIPS codes as number.[1] This results in the three-digit county FIPS code sometimes containing less than three digits (e.g., "1" as opposed to "001"). When combining state and county FIPS codes, this can lead to mis-located state FIPS code (e.g., `paste0("42","1")="421"` while it is supposed to be "42001").

To deal with this issue, the updated version change the way combined FIPS codes are generated in the *Create Variables* section of *002-flatten_data.R*. Rather than combining character variables, I treat them as numeric variables and combine them by $1000 \times FIPS\_state + FIPS\_county$.

Note that because of this operation, the resulting dataset has origin and destination variables as numeric rather than character. If one prefers to transfer them to character variables, they should be aware that they may need to imput the 5th digit for those in states with one digit state FIPS code (e.g., "01001" will be shown as 1001 in current dataset).

This problem only happened to data of Year 1998.

Correct a coding error in raw data in Year 2003

There is a coding error in cell B9 of a data file[2]. The county FIPS is supposed to be "000" (total migrants of the whole state) but was instead input as "17". As a result, there exists an artificial entrywith origin=99999 (un-identified) and destination=1717. 1717 is a non-existing FIPS code, and this row of data only has non-zero entry in Year 2003. Nevertheless, since the migrant size there is $459,424$, it could bring an artificial elevation in the total number of migrants if this artificial data entry was not excluded. The updated script *002-flatten_data.R* includes a line of hard-coded correction in the section of *1995-2003*.

[1] An example can be found at: */MigData/1998to1999/1998to1999CountyMigration/1998to1999CountyMigrationInflow/co989pai.xls*

[2] */MigData/2003to2004/2003to2004CountyMigration/2003to2004CountyMigrationInflow/co0304ILi.xls*

Resolve errors in reading and processing excel files

The function `read_excel()` used in reading excel files requires the `readxl` package, which is now included in *000-libraries.R*.

Probably due to pacakge version difference, when using `readxl version 1.4.3`, the column names were generated as "...1" rather than "X___1" (number 1 indicating the column number is 1 for example), which creates incompatibility with existing commands and causing errors. To minimize modifications of the data, the updated *002-flatten_data.R* include an operation to change all column names into the format of "X___1" for every year from 1992 to 2010 before further processing the data. This does not change the output data, but resolve errors of running existing scripts with the updated package.

Resolve errors in processing data of Years 1990,1991

The previous script generated errors when processing data of Years 1990,1991, which was resolved in this updated version. This update does not change the output data.

To explain what happened turns out to be non-trivial, so readers can ignore this section if they get bored. But to keep a record, it all comes from the complicated data structure of data files for Years 1990-1991. Taking inflow files as an example, each destination's data starts with a "master" row indicating the destination, followed by rows indicating migration flows of each origin. They also include rows that summarize the total number of in-state migrant, migration towards Northeast region, or migration abroad etc for each county. Vice versa for the outflow. To process this data, the summary rows were removed, and the previous script did that by removing any rows that had keywords including "foreign." This creates a problem: there are also chunks of data that summarize international migration, where the master row contains the keyword "foreign". Removing that master row generated error messages because the algorithm did not know what's the destination of the migration flows for the following rows. To resolve this we have to preserve that kind of master rows, and I do that by preserving any rows that contains "/", as a way to identify them.

Correct unidentified flows calculation in Years 1990,1991

Another issue for data of Years 1990,1991 is the underestimation of the unidentified migration flow out of counties (coded as destination FIPS 99999), due to the ignorance of migration towards APO / FPO Zip Code (oversea military bases or fleets). This influences only 1 county by 12 migrants in Yr 1990 from Honululu, HI, the only county that sent more than 10 migrants to be oversea military personnel to be identified in IRS data. But 1,117 counties in Yr 1991 were impacted. This huge difference makes sense as the Gulf War took place in 1991. The issue can also influence the calculation of total migrant counts of those counties and years using the previous data. Total migrants in Year 1990 was underestimated by 12, and total migrants in Year 1991 was underestimated by 186,985.

What happened in the algorithm is again complicated, and here is for the record: the way those unidentified flows are calculated is by substracting identified inter-county migration flows from the total migrant count of each county. So when migration flows to/from APO/FPO are not excluded in that calculation, the unidentified flows would be underestimated. While the previous script removed other international flows with keyword "Foreign", "57 005 APO / FPO Zip Code" were remained. This was not a problem for inflow data, because they deleted any rows with non-identified state FIPS code such as "57" that refers to international migration in IRS data. However, for outflow data, the previous script made a mistake of not filtering data based on destination, but still filtering it based on origin, which cased the problem. The updated script fixed the issue by filtering using state FIPS code for both origin and destination for both inflows and outflows: `data_table <- data_table[org_state %in% us_states$FIPS_CODE & des_state %in% us_states$FIPS_CODE]`.

Clarification of the meaning of unidentified flows

The processed data uses FIPS=99999 for destination when the migration flow out of a county has an unknown destination (vice versa for origin). By adding this number with all other identified inter-county migration flows, users will get the total number of migrants from or to a county measured by IRS.

It should be noted that the total migrant number collected by IRS includes international migration. This means that the unidentified flows contain not only inter-county migration flows with less than 10 migrants (reported without identified origin/destination for privacy reason), as noted in Hauer & Byars (2019), but also international migration flows. IRS does report the scale of international migration in their raw data, so it is possible for users to modify the scripts and further distinguish domestic and international migration flows.

Concluding notes

Processing raw data is a necessary but sometimes/usually tedious step in research. Therefore it is great services of Hauer and Byars in publishing this dataset and their scripts for replication.

As described in their article, processing this large dataset with varying data structures takes tons of work, so it is absolutely understandable that the previous version was imperfect. In the meantime, it is also likely that there are still remaining issues after this update. Therefore, I would request users to cite not only this repository but also Hauer & Byars (2019) for their hard work in making this update possible in the first place. I would also invite future users to further examine the data and develop this project. Thank you for contributing to open science.