Enhancing reproducibility with the IPUMS API and the ipumsr package

Derek Burk, Dan Ehrlich, & Kara Fisher

4/11/2022

Who we are

Derek Burk, PhD

Sociology

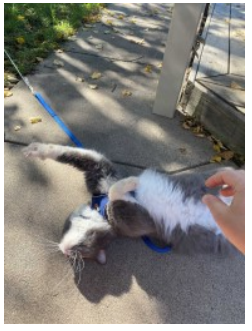
Dan Ehrlich, MA

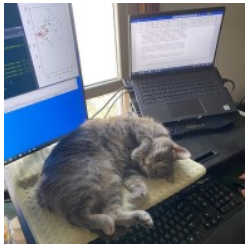
Anthropology

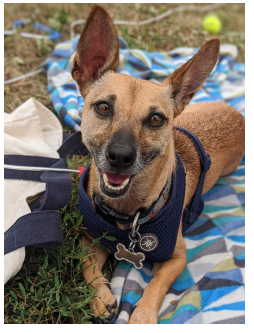
Kara Fisher, MPH

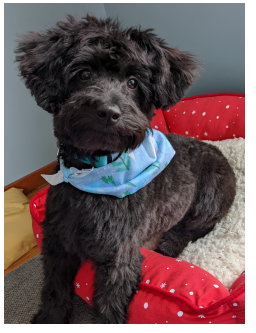
Public Health

Who we are









INSTITUTE FOR
SOCIAL RESEARCH & DATA INNOVATION

IPUMS MPC MN RDC LIFE COURSE CENTER

Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API
7. IPUMS API use cases
8. Q & A

Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API
7. IPUMS API use cases
8. Q & A

What is IPUMS?

IPUMS is **data**

from censuses and surveys around the world,

harmonized across space and time,

thoroughly documented,

and available for free at ipums.org

Harmonization

Variable harmonization: marital status

Harmonized		Input		
Code	Label	Bangladesh 2011	Mexico 1970	Kenya 1999
1 0 0	Single	1 = Unmarried	8 = Single	1 = Never married
2 0 0	Married or in union	2 = Married		
2 1 0	Married, formally			
2 1 1	Civil		2 = Married, civil	
2 1 2	Religious		3 = Married, religious	
2 1 3	Civil and religious		1 = Married, civil & relig	
2 1 4	Monogamous			2 = Monogamous
2 1 5	Polygamous			3 = Polygamous
2 2 0	Consensual union		4 = Consensual union	
3 0 0	Divorced or separated	4 = Divrc or separated		
3 1 0	Separated		7 = Separated	6 = Separated
3 2 0	Divorced		6 = Divorced	5 = Divorced
4 0 0	Widowed	3 = Widowed	5 = Widowed	4 = Widowed

- U.S. Census and American Community Survey **microdata** from 1850 to the present.
- 180,755,919 unique person records from decennial census and American Community Survey.
- 191,983,898 historical person records from full count decennial census from 1850-1940 (1890 census lost due to fire).
- https://usa.ipums.org/usa/

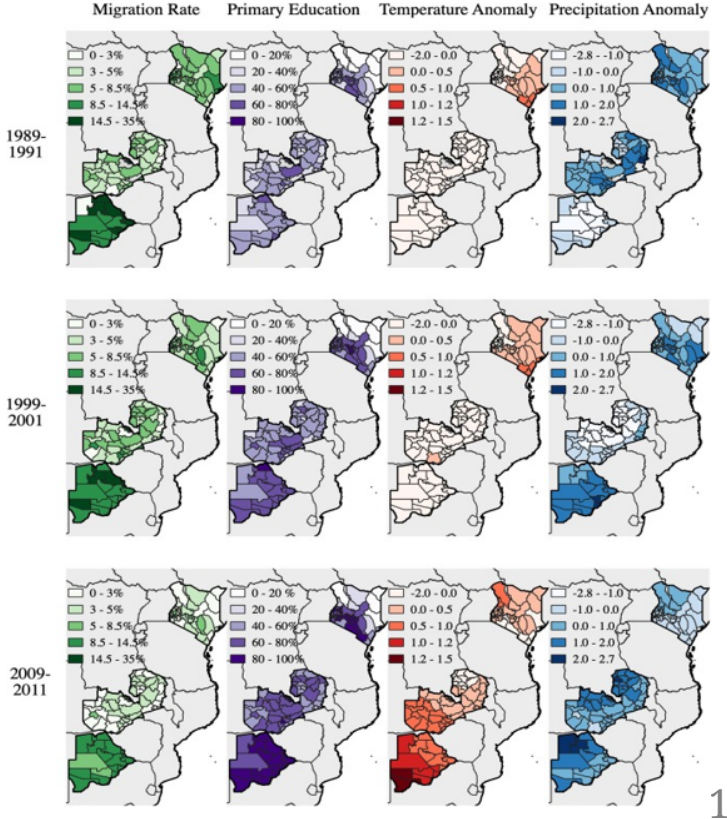
- Current Population Survey **microdata** from 1962 to the present.
- Monthly labor force surveys and supplements.
- https://cps.ipums.org/cps/

- Health **survey** data from the National Health Interview Survey (NHIS) from the 1960s to the present and the Medical Expenditure Panel Survey (MEPS) from 1996.
- Supplements on cost of healthcare.
- https://healthsurveys.ipums.org/

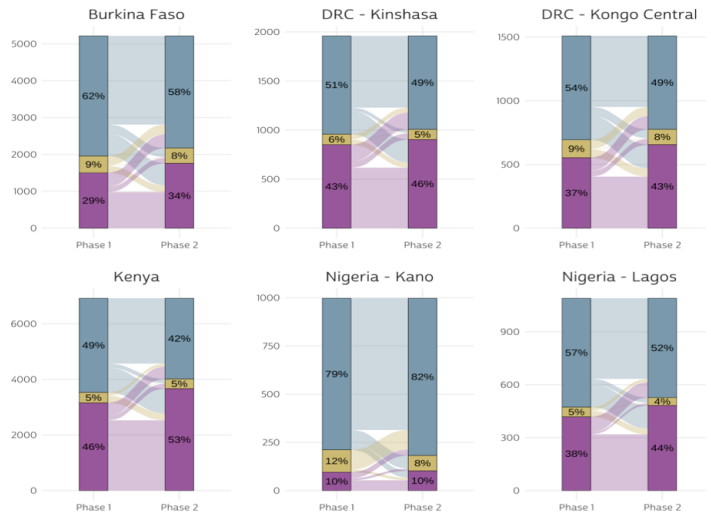
- Scientists and Engineers Statistical Data System (SESTAT), the leading surveys for studying the science and engineering (STEM) workforce in the United States
- Data from the National Surveys of College Graduates (NSCG), Recent College Graduates (NSRCG) and Doctorate Recipients (SDR) are integrated from 1993 to the present.
- https://highered.ipums.org/highered/

- Census **microdata** covering 103 countries from 1960 to the present
- International historic **microdata** from the 19th and early 20th centuries available for some samples.
- Labor Force surveys provide high resolution **microdata** about work conditions
    - Administered quarterly (usually) with records going back at least 10 years (usually)
    - Currently available for Italy (2011-2020), Spain (2005-2020), and Mexico (2005-2020)
- https://international.ipums.org/international/

- “Climate-Induced migration and unemployment in middle-income Africa”
- Valerie Mueller, Clark Gray, and Douglas Hopping



- Demographic and Health Surveys (DHS) provide integrated **microdata** for analysis across time and space.
    - From the 1980s to the present.
    - Covering Africa and South Asia
- Performance Monitoring for Action (PMA) surveys
    - Focus on fertility, contraception, hygiene, and health
    - Administered frequently to monitor trends in select high-fertility countries.
    - https://globalhealth.ipums.org/

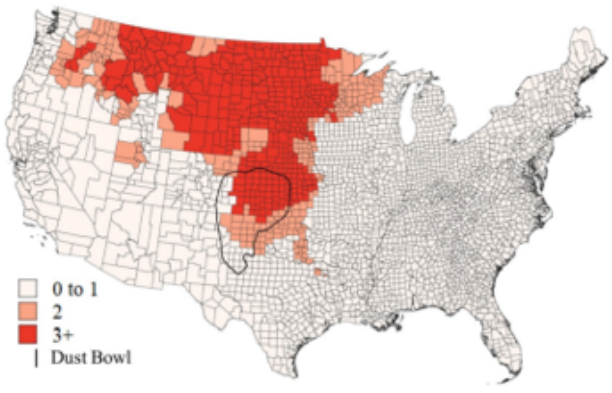


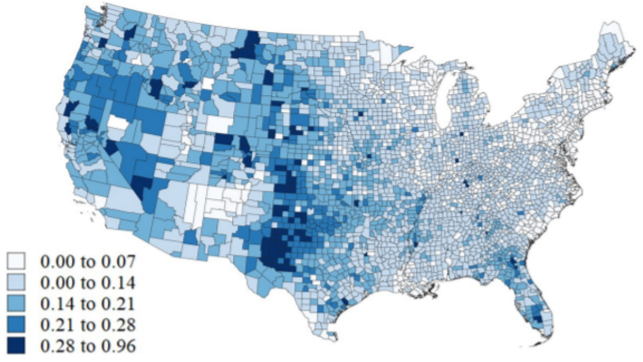
- https://tech.popdata.org/pma-data-hub/

- **NHGIS** Shapefiles for all levels of US geography including tracts, from 1790 to the present
- https://www.nhgis.org/
- **IHGIS** Shapefiles for admin level 2
- https://ihgis.ipums.org/
- Summary tables and time series of population, housing, agriculture, and economic data

- “Migrant Selection and Sorting during the Great American Drought.”
- Christopher Sichko

Figure 1: Number of drought years from 1935 to 1939





(a) < 8th grade education

- Historical and contemporary time use data from 1965 to the present.
- Extensive time diary data from respondents in the US and 7 other countries.
- https://timeuse.ipums.org/

- Nathan Yau of Flowing Data
- http://flowingdata.com/projects/2015/simulation/

So what is IPUMS?

- IPUMS is **a lot** of data
- Individual-level microdata
- Summarized tabular data
- GIS shapefiles
- Consistent and extensively documented **metadata**

So what is IPUMS?

- IPUMS is **a lot** of data
- Individual-level microdata
- Summarized tabular data
- GIS shapefiles
- Consistent and extensively documented **metadata**
- *How can I work with all this IPUMS data?*

Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API
7. IPUMS API use cases
8. Q & A

What is ipumsr?

- R package developed by Greg Freedman Ellis
- Released in 2017
- Over 100,000 CRAN downloads
- Includes functions for
  - Reading IPUMS data
  - Exploring and manipulating IPUMS metadata
  - **SOON**: Interacting with the IPUMS API

Why use ipumsr?

- One package for IPUMS microdata, aggregate data, and geography
- Specialized functions for viewing and manipulating IPUMS metadata
- Bundled how-to guides (vignettes)
- Potential to add more features (e.g. API support); let us know what you want!
  - File an issue at https://github.com/ipums/ipumsr/issues
  - Email ipums+cran@umn.edu

To run the code in this presentation

Install R packages (as needed)

```
install.packages("ipumsr")

## Tidyverse
install.packages("dplyr")
install.packages("ggplot2")
install.packages("stringr")
install.packages("purrr")

## HTML tables
install.packages("DT")

## GitHub helper functions
install.packages("usethis")
```

Load R packages (each time)

```
library(ipumsr)

## Tidyverse
library(dplyr)
library(ggplot2)
library(stringr)
library(purrr)

## HTML tables
library(DT)

## GitHub helper functions
library(usethis)
```

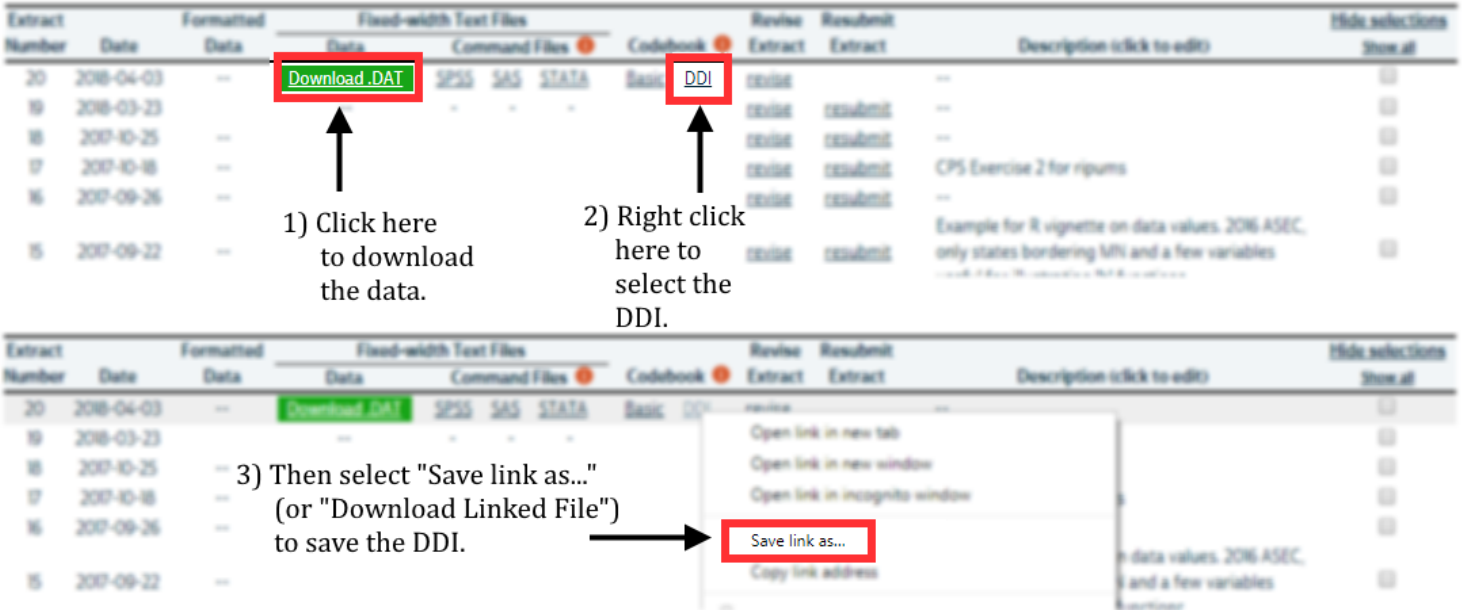
Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API
7. IPUMS API use cases
8. Q & A

Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API
7. IPUMS API use cases
8. Q & A

Downloading your data extract



- You must download both the data and DDI codebook
- Save both files in the same folder

Downloading your data extract

- Optional: "R" link contains code to read in your data with ipumsr

Extract Number	Date	Formatted Data	Fixed-width Text Files						Revise Extract	Resubmit Extract	Description (click to edit)
			Data	Command Files				Codebook			
59	2021-09-27	--	Download .DAT	SPSS	SAS	Stata	R	Basic DDI			Family structure Saint Lucia 1980

Read in the data

- Using functions `read_ipums_ddi()` and `read_ipums_micro()`

```
ddi <- read_ipums_ddi("usa_00013.xml")

data <- read_ipums_micro(ddi)
```

- Note: supply the codebook, *not* the data file, to `read_ipums_micro()`

The data file is just raw ingredients

```
20152015020000000100000000000740000003200007033000099999990000010000000
20152015020000000200000000002210000006700007031432000120000300010000000
20152015020000000200000000002210000006700007031432000120000300020000000
20152015020000000200000000002210000006700007031432000120000300030000000
20152015020000000300000000003100000006000007031048000230000100010000000
20152015020000000300000000003100000006000007031048000230000100020000000
20152015020000000300000000003100000006000007031048000230000100030000000
20152015020000000300000000003100000006000007031048000230000100040000000
20152015020000000300000000003100000006000007031048000230000100050000000
20152015020000000400000000010870000011000007031084000436000100010000001
```

The DDI codebook is the recipe

```
names(ddi)
#>  [1] "file_name"         "file_path"
#>  [3] "file_type"         "ipums_project"
#>  [5] "extract_date"      "extract_notes"
#>  [7] "rectypes"          "rectype_idvar"
#>  [9] "rectypes_keyvars" "var_info"
#> [11] "conditions"        "citation"
#> [13] "file_encoding"
```

```
ddi$file_name
#> [1] "usa_00104.dat"
```

The DDI codebook is the recipe

```
ddi$var_info
#> # A tibble: 35 x 10
#>    var_name var_label                                          var_c
#>    <chr>    <chr>                                              <chr>
#>  1 YEAR     "Census year"                                      "YEA
#>  2 SAMPLE   "IPUMS sample identifier"                          "SAM
#>  3 SERIAL   "Household serial number"                         "SER
#>  4 CBSERIAL "Original Census Bureau household serial number" "CBS
#>  5 HHWT     "Household weight"                                 "HHW
#>  6 CPI99    "CPI-U adjustment factor to 1999 dollars"          "CPI
#>  7 GQ       "Group quarters status"                            "GQ
#>  8 COSTELEC "Annual electricity cost"                          "COS
#>  9 HHINCOME "Total household income "                          "HHI
#> 10 VACANCY  "Vacancy status"                                   "VAC
#> # ... with 25 more rows
```

Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API
7. IPUMS API use cases
8. Q & A

What's in my extract again?

We can print the names of our variables:

```
names(data)
#>  [1] "YEAR"      "SAMPLE"    "SERIAL"
#>  [4] "CBSERIAL"  "HHWT"      "CPI99"
#>  [7] "GQ"        "COSTELEC"  "HHINCOME"
#> [10] "VACANCY"   "CINETHH"   "PERNUM"
#> [13] "PERWT"     "MOMLOC"    "POPLOC"
#> [16] "RELATE"    "RELATED"   "SEX"
#> [19] "AGE"       "MARRINYR"  "DIVINYR"
#> [22] "WIDINYR"   "FERTYR"    "RACE"
#> [25] "RACED"     "HISPAN"    "HISPAND"
#> [28] "SPEAKENG"  "EDUC"      "EDUCD"
#> [31] "EMPSTAT"   "EMPSTATD"  "INCTOT"
#> [34] "MIGRATE1"  "MIGRATE1D"
```

But often variable names aren't self-explanatory.

Let's leverage that attached metadata!

Available metadata

Variable labels and descriptions:

```
ipums_var_label(ddi, MIGRATE1)
#> [1] "Migration status, 1 year [general version]"

ipums_var_desc(ddi, MIGRATE1) %>%
  substr(1,450) %>%
  strwrap(60)
#> [1] "MIGRATE1 reports whether the person had changed residence"
#> [2] "since a reference point 1 year ago. Specifically,"
#> [3] "individuals age 1+ were asked if they had lived in the"
#> [4] "\"same house\" (non-movers) or a \"different house\" (movers,
#> [5] "one year earlier. Persons who had moved were to indicate"
#> [6] "the foreign country or the state, county, and place of"
#> [7] "their normal residence during the reference year. Migration"
#> [8] "data were collected only for sample-line persons in 1950."
```

Available metadata

Value labels:

```
ipums_val_labels(ddi, MIGRATE1)
#> # A tibble: 6 x 2
#>     val lbl
#>   <dbl> <chr>
#> 1     0 N/A
#> 2     1 Same house
#> 3     2 Moved within state
#> 4     3 Moved between states
#> 5     4 Abroad one year ago
#> 6     9 Unknown
```

An interactive view of metadata

```
ipums_view(ddi)
```

+ MIGRATE1

Migration status, 1 year [general version]

Variable Description

MIGRATE1 reports whether the person had changed residence since a reference point 1 year ago. Specifically, individuals age 1+ were asked if they had lived in the "same house" (non-movers) or a "different house" (movers) one year earlier. Persons who had moved were to indicate the foreign country or the state, county, and place of their normal residence during the reference year. Migration data were collected only for sample-line persons in 1950.

The category "Same house" includes all eligible persons who did not move since the reference year, as well as those who had moved but by the enumeration or survey date had returned to their earlier residence. The category "Different house" includes persons who lived in a different house in the reference year. For 1950, movers (those who reported living in a different house in the reference year) are further subdivided according to type of move (e.g., within the county or across state lines). The ACS and the PRCS report only same/different residence and identifies those previously living abroad.

Therefore, for the ACS/PRCS samples, MIGRATE1 uses information contained in the IPUMS variable MIGPLAC1 and compatible PUMAs of migration and PUMAs of residence to indicate whether movers migrated between states or within the same state (the same levels of detail in the 1950 classification.). For movers who migrated between states, a detailed version of MIGRATE1 indicates whether they moved between contiguous or non-contiguous states. For movers who migrated within the same state, detailed MIGRATE1 indicates whether they moved within or between PUMAs.

More details

Value Labels

Coding Instructions

N/A

Labelled Values

Show 10 entries Search:

val	lbl
0	N/A
1	Same house
2	Moved within state
3	Moved between states
4	Abroad one year ago
9	Unknown

Showing 1 to 6 of 6 entries Previous 1 Next

How do I manipulate metadata?

Wrangling value labels

- R doesn't natively support value labels like other statistical packages do
- `ipumsr` uses `haven::labelled()` objects to preserve values and labels, but these objects can be tricky to work with
- `ipumsr` helper functions allow you to leverage info from values and labels
- When you are finished manipulating variables, convert to factor using `as_factor()` or convert to character or numeric vector

Using ipumsr label helper functions

haven::labelled columns at a glance

- Let's look at the detailed education variable

```
ipums_val_labels(data$EDUCD)
#> # A tibble: 44 x 2
#>      val lbl
#>    <int> <chr>
#>  1     0 N/A or no schooling
#>  2     1 N/A
#>  3     2 No schooling completed
#>  4    10 Nursery school to grade 4
#>  5    11 Nursery school, preschool
#>  6    12 Kindergarten
#>  7    13 Grade 1, 2, 3, or 4
#>  8    14 Grade 1
#>  9    15 Grade 2
#> 10    16 Grade 3
#> # ... with 34 more rows
```

lbl_collapse()

- `lbl_collapse()` allows you to take advantage of the hierarchical structure of value labels

```
ipums_val_labels(data$EDUCD)
#> # A tibble: 44 x 2
#>      val lbl
#>    <int> <chr>
#>  1     0 N/A or no schooling
#>  2     1 N/A
#>  3     2 No schooling completed
#>  4    10 Nursery school to grade 4
#>  5    11 Nursery school, preschool
#>  6    12 Kindergarten
#>  7    13 Grade 1, 2, 3, or 4
#>  8    14 Grade 1
#>  9    15 Grade 2
#> 10    16 Grade 3
#> # ... with 34 more rows
```

lbl_collapse()

- `lbl_collapse()` allows you to take advantage of the hierarchical structure of value labels

```
data$EDUCD2 <- lbl_collapse(data$EDUCD, ~.val %/% 10) %>%
  as_factor(ordered = TRUE)
```

lbl_collapse()

- All `ipumsr` helper functions allow you to use `.val` or `.lbl` to refer to values or value labels

```
data$EDUCD2 <- lbl_collapse(data$EDUCD, ~.val %/% 10) %>%
  as_factor(ordered = TRUE)
```

lbl_collapse()

- In this case, we want to collapse the last digit

```
data$EDUCD2 <- lbl_collapse(data$EDUCD, ~.val %/% 10) %>%
  as_factor(ordered = TRUE)
```

lbl_collapse()

```
data$EDUCD2 <- lbl_collapse(data$EDUCD, ~.val %/% 10) %>%
  as_factor(ordered = TRUE)
```

...

before ([val] label)	after	count
[0] N/A or no schooling	N/A or no schooling	0
[1] N/A	N/A or no schooling	2741
[2] No schooling completed	N/A or no schooling	4850
[10] Nursery school to grade 4	Nursery school to grade 4	0

Still too detailed for a graph

```
data$EDUCD %>%
  lbl_collapse(~.val %/% 10) %>%
  ipums_val_labels()
#> # A tibble: 13 x 2
#>      val lbl
#>    <dbl> <chr>
#>  1     0 N/A or no schooling
#>  2     1 Nursery school to grade 4
#>  3     2 Grade 5, 6, 7, or 8
#>  4     3 Grade 9
#>  5     4 Grade 10
#>  6     5 Grade 11
#>  7     6 Grade 12
#>  8     7 1 year of college
#>  9     8 2 years of college
#> 10     9 3 years of college
#> 11    10 4 years of college
#> 12    11 5+ years of college
#> 13    99 Missing
```

lbl_relabel()

- Maybe the education variable is still too specific.

```
college_regex <- "^[123] year(s)? of college$"

data$EDUCD3 <- data$EDUCD %>%
  lbl_collapse(~.val %/% 10) %>%
  lbl_relabel(
    lbl(2, "Less than High School") ~.val > 0 & .val < 6,
    lbl(3, "High school") ~.lbl == "Grade 12",
    lbl(4, "Some college") ~str_detect(.lbl, college_regex),
    lbl(5, "College or more") ~.val %in% c(10, 11)
  ) %>%
  as_factor()
```

lbl_relabel()

- Maybe the education variable is still too specific.

```
college_regex <- "^[123] year(s)? of college$"

data$EDUCD3 <- data$EDUCD %>%
  lbl_collapse(~.val %/% 10) %>%
  lbl_relabel(
    lbl(2, "Less than High School") ~.val > 0 & .val < 6,
    lbl(3, "High school") ~.lbl == "Grade 12",
    lbl(4, "Some college") ~str_detect(.lbl, college_regex),
    lbl(5, "College or more") ~.val %in% c(10, 11)
  ) %>%
  as_factor()
```

lbl_relabel()

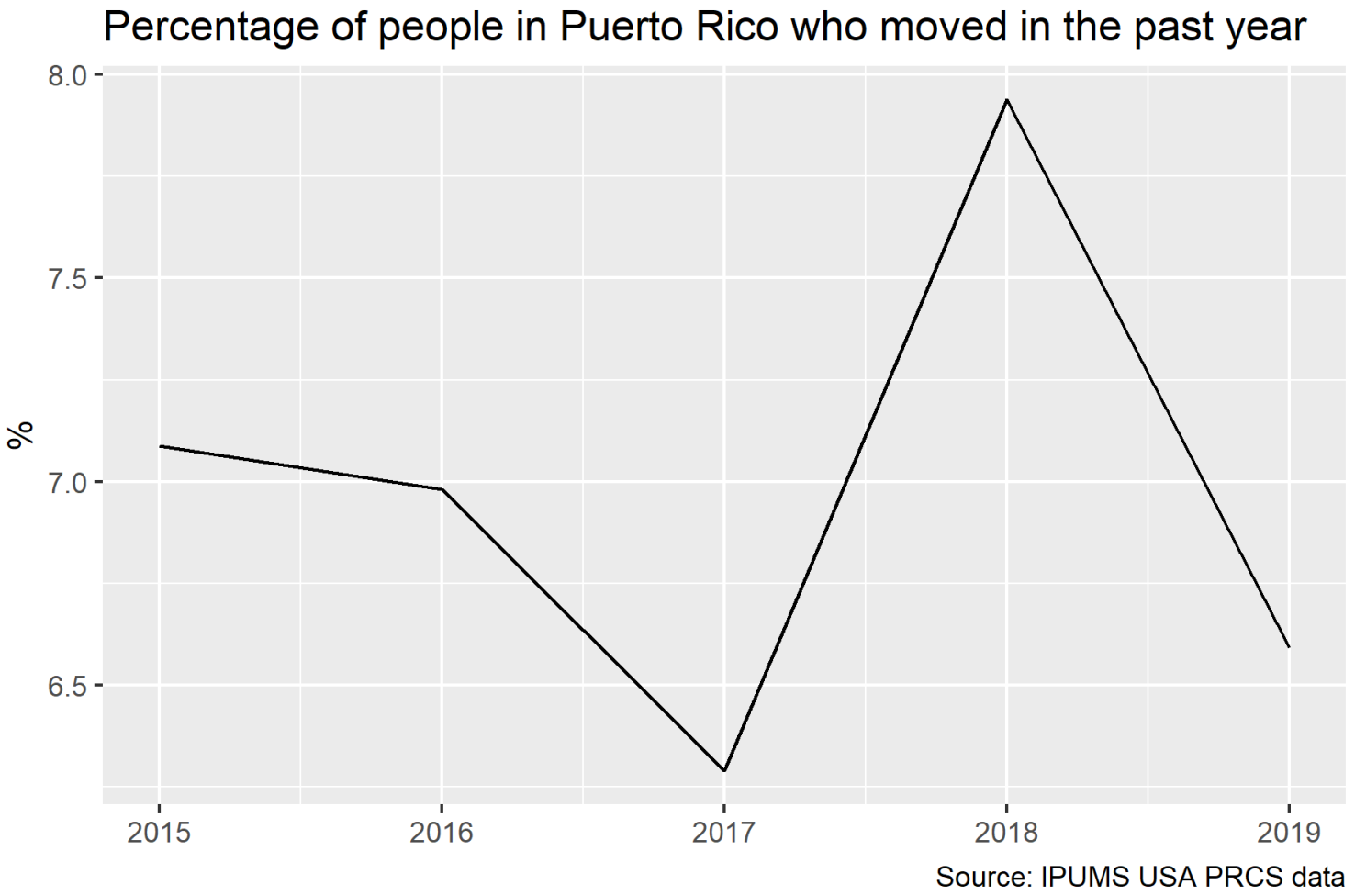
- Maybe the education variable is still too specific.

```
college_regex <- "^[123] year(s)? of college$"

data$EDUCD3 <- data$EDUCD %>%
  lbl_collapse(~.val %/% 10) %>%
  lbl_relabel(
    lbl(2, "Less than High School") ~.val > 0 & .val < 6,
    lbl(3, "High school") ~.lbl == "Grade 12",
    lbl(4, "Some college") ~str_detect(.lbl, college_regex),
    lbl(5, "College or more") ~.val %in% c(10, 11)
  ) %>%
  as_factor()
```

lbl_relabel()

```
levels(data$EDUCD3)
```

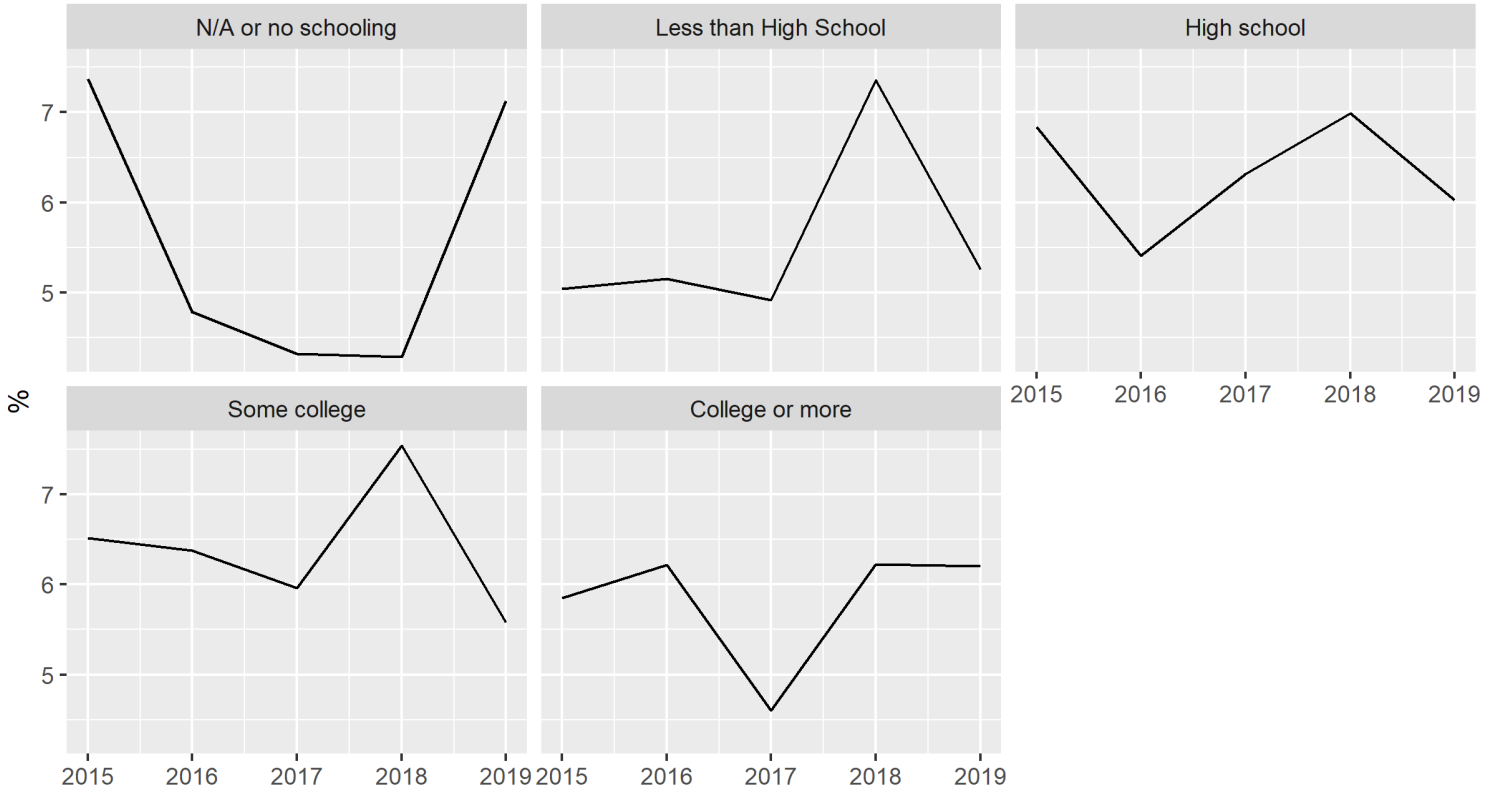
```
#> [1] "N/A or no schooling"
#> [2] "Less than High School"
#> [3] "High school"
#> [4] "Some college"
#> [5] "College or more"
#> [6] "Missing"
```



Note: To estimate confidence intervals, use variables REPWT or REPTWP

Percentage of people in Puerto Rico who moved in the past year

Persons age 25 and older



Source: IPUMS USA PRCS data

Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API
7. IPUMS API use cases
8. Q & A

API Timeline

- Currently in beta testing for IPUMS USA
- IPUMS USA public launch expected mid-2022
- Interested in becoming a beta tester? Email ipums+api@umn.edu

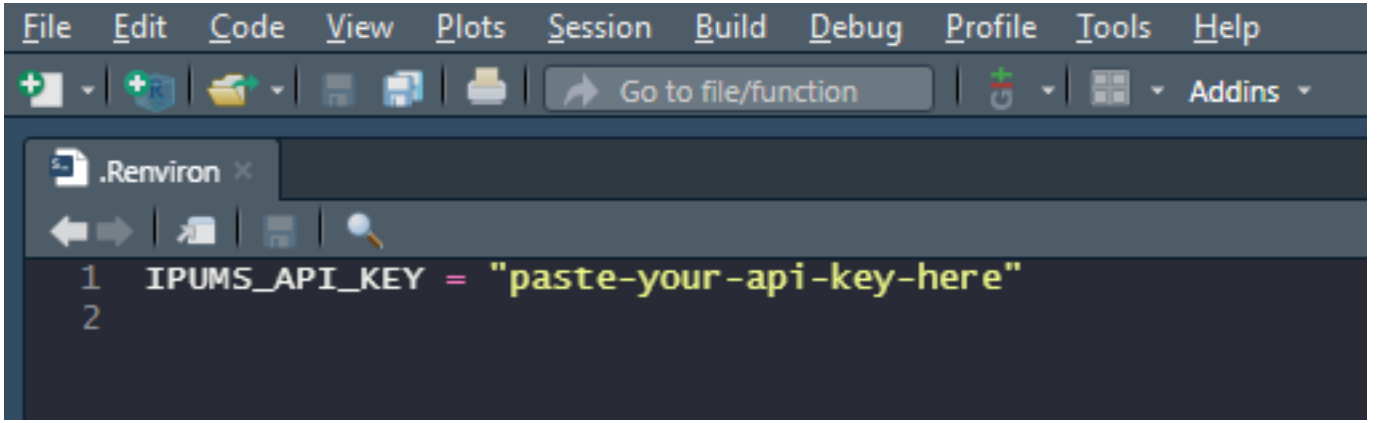
What can I do with the IPUMS USA API?

- Define and submit extract requests
- Check extract status or "wait" for an extract to finish
- Download completed extracts
- Get info on past extracts
- Share extract definitions

What can't I do with the IPUMS USA API?

- Bypass the extract system entirely
- Explore what data are available
- Use all features of the extract system (at least not right away)

How to use your IPUMS USA API key



Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API
7. IPUMS API use cases
8. Q & A

Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API

7a. API use case 1: Revise your extract

8. Q & A

Add HHINCOME to our extract

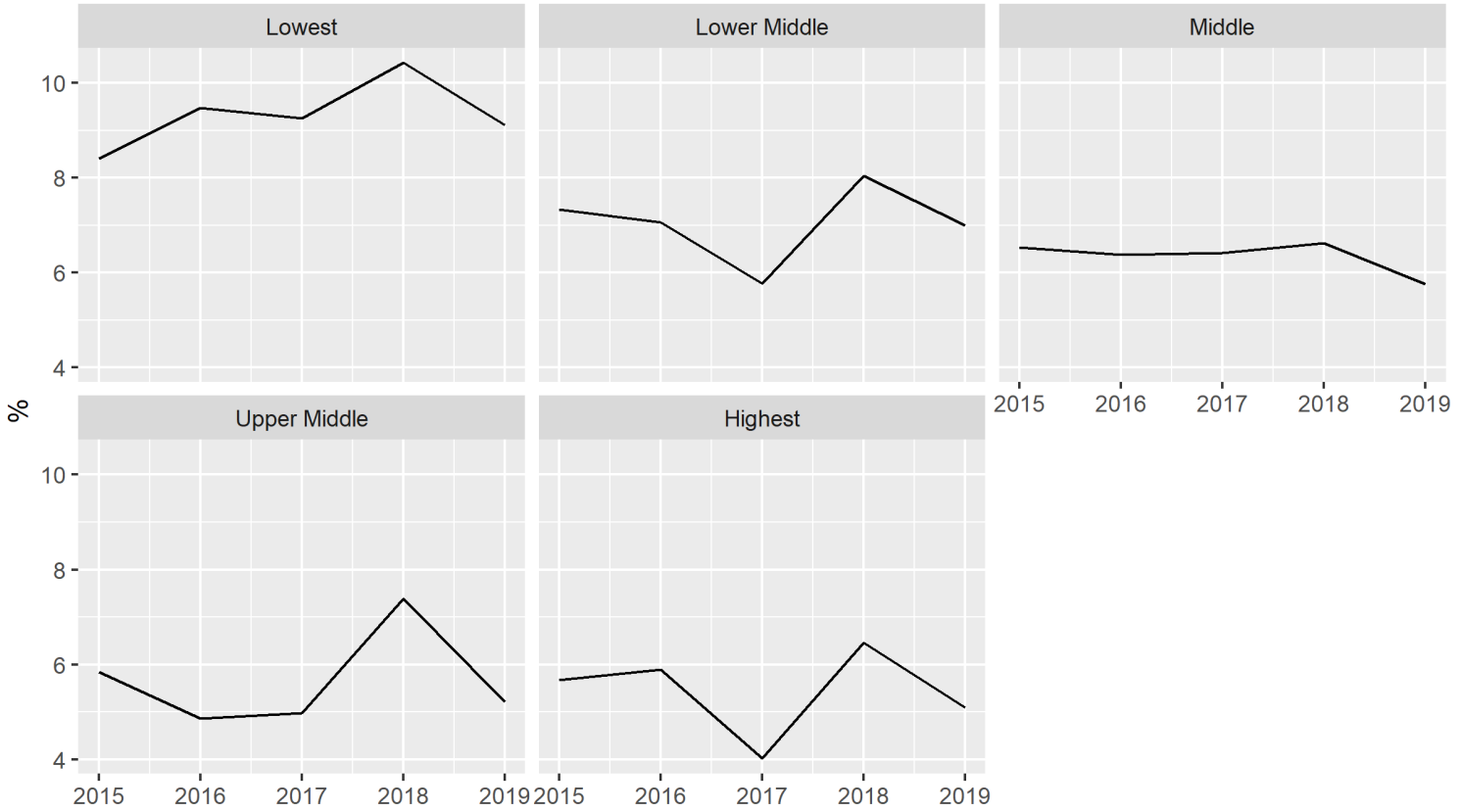
```
# Pull down definition of last extract
last_extract <- get_last_extract_info("usa")

# Add HHINCOME, resubmit, and download
ddi <- last_extract %>%
  revise_extract_micro(vars_to_add = "HHINCOME") %>%
  submit_extract() %>%
  wait_for_extract() %>%
  download_extract() %>%
  read_ipums_ddi()

data <- read_ipums_micro(ddi)
```

Update your script with new filename

```
# Before
ddi <- read_ipums_ddi("usa_00013.xml")
data <- read_ipums_micro(ddi)

# After
ddi <- read_ipums_ddi("usa_00014.xml")
data <- read_ipums_micro(ddi)
```

Percentage of people who moved in the past year, 2015-2019

Grouped by household income quintile



How to create the extract from scratch

```
my_extract <- define_extract_micro(
  "usa",
  description = "Puerto Rico migration extract",
  samples = c("us2015b", "us2016b", "us2017b", "us2018b", "us2019b"),
  variables = c("AGE", "SEX", "EDUC", "MIGRATE1", "HHINCOME")
)

ddi <- my_extract %>%
  submit_extract() %>%
  wait_for_extract() %>%
  download_extract() %>%
  read_ipums_ddi()

data <- read_ipums_micro(ddi)
```

Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API
7a. API use case 1: Revise your extract
7b. API use case 2: Share your extract
8. Q & A

Share your extract (definition)

```
extract_to_share <- get_last_extract_info("usa")

# Or if you have the extract number:
extract_to_share <- get_extract_info("usa:14")

save_extract_as_json(
  extract_to_share,
  file = "prcs_migration_extract.json"
)
```

- Then share "prcs_migration_extract.json" with collaborators

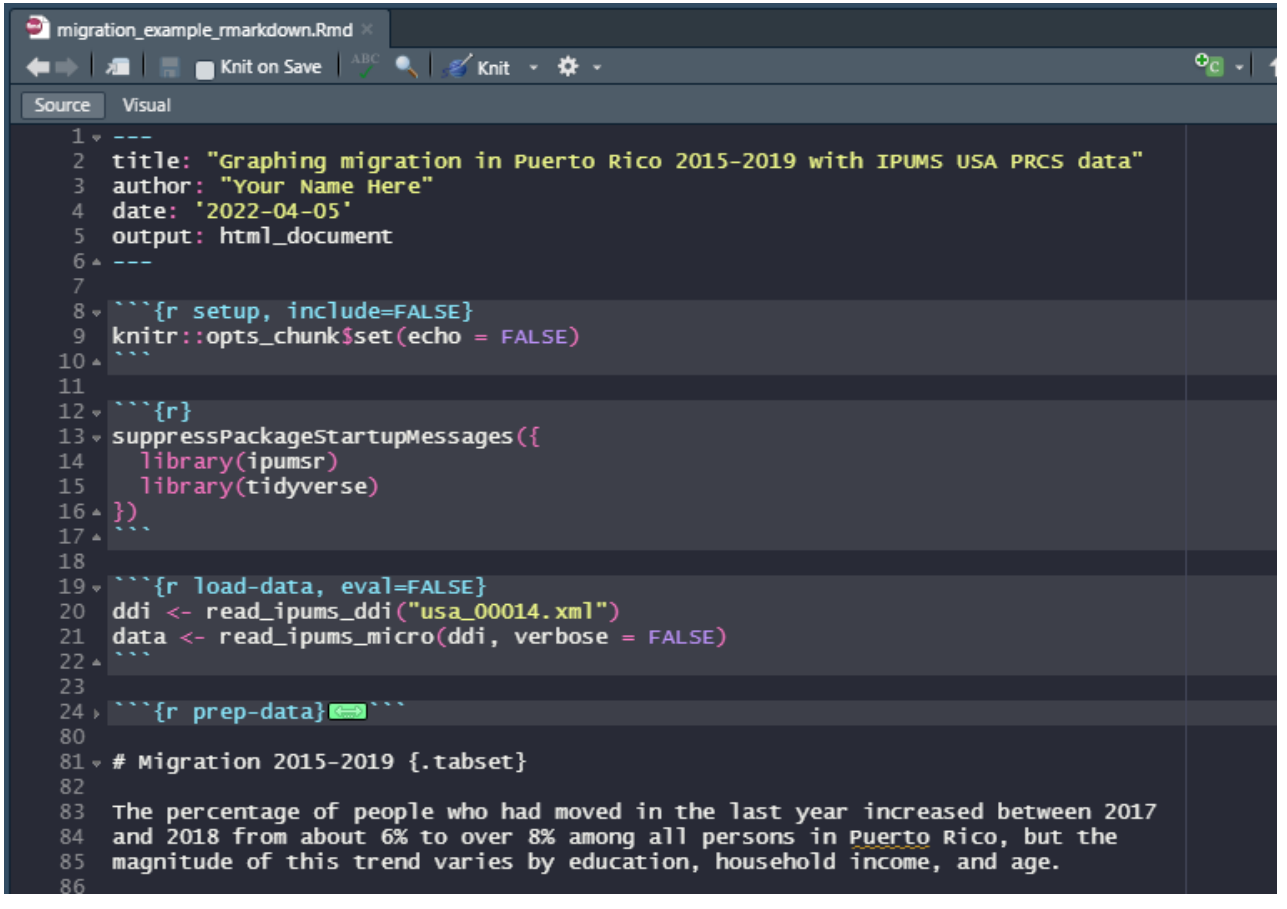
Clone a shared extract

```
shared_extract_definition <- define_extract_from_json(
  "prcs_migration_extract.json",
  collection = "usa"
)

ddi <- shared_extract_definition %>%
  submit_extract() %>%
  wait_for_extract() %>%
  download_extract() %>%
  read_ipums_ddi()

data <- read_ipums_micro(ddi)
```

Overview

1. What is IPUMS?
2. What is ipumsr, and why use it?
3. How to create a data extract
4. Reading data into R
5. Exploring and manipulating metadata
6. Intro to the IPUMS USA API
7a. API use case 1: Revise your extract
7b. API use case 2: Share your extract
7c. API use case 3: Share your analysis
8. Q & A

Save analysis as RMarkdown



Graphing migration in Puerto Rico 2 ×

file:///C:/Users/derek/Documents/ipumsr-psu-api-workshop/migration_example_rm

Graphing migration in Puerto Rico 2015-2019 with IPUMS PRCS data

Your Name Here

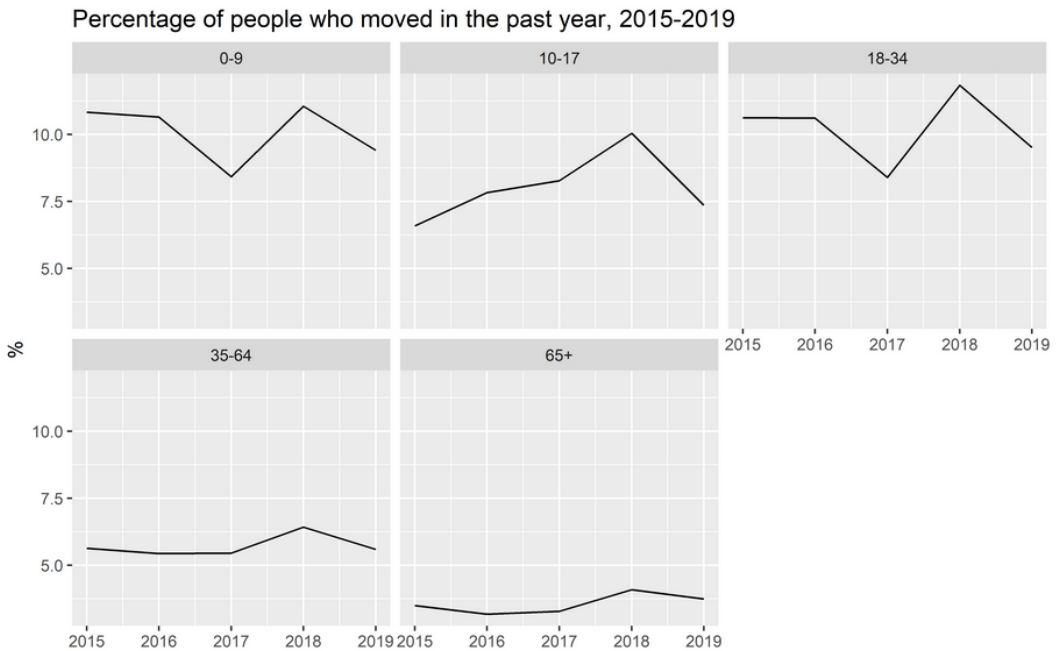
2022-04-05

Migration 2015-2019

The percentage of people who had moved in the last year increased between 2017 and 2018 from about 6% to over 8% among all persons in Puerto Rico, but the magnitude of this trend varies by education, household income, and age.

Note: These graphs show trends in point estimates from sample data, without displaying estimates of sampling error. Differences over time or across groups may not be statistically significant. To calculate confidence intervals for point estimates, follow the IPUMS USA instructions for using replicate weights.

Overall | By education | By household income | By age



Instead of hard-coding the extract number...

This only works for you:

```
ddi <- read_ipums_ddi("usa_00014.xml")
data <- read_ipums_micro(ddi)
```

...download the data if not available locally!

```
if (!file.exists("prcs_migration_extract.xml")) {
  # Load extract definition from JSON
  prcs_migration_extract <- define_extract_from_json(
    "prcs_migration_extract.json",
    "usa"
  )
  # Submit, wait for, and download extract
  ddi_filename <- submit_extract(prcs_migration_extract) %>%
    wait_for_extract() %>%
    download_extract() %>%
    basename()
  # Infer data file name from DDI file name
  data_filename <- str_replace(ddi_filename, "\\.xml$", ".dat.gz")
  # Standardize DDI and data file names
  file.rename(ddi_filename, "prcs_migration_extract.xml")
  file.rename(data_filename, "prcs_migration_extract.dat.gz")
}
```

Then hard-code the renamed files

Now this will work for anyone!

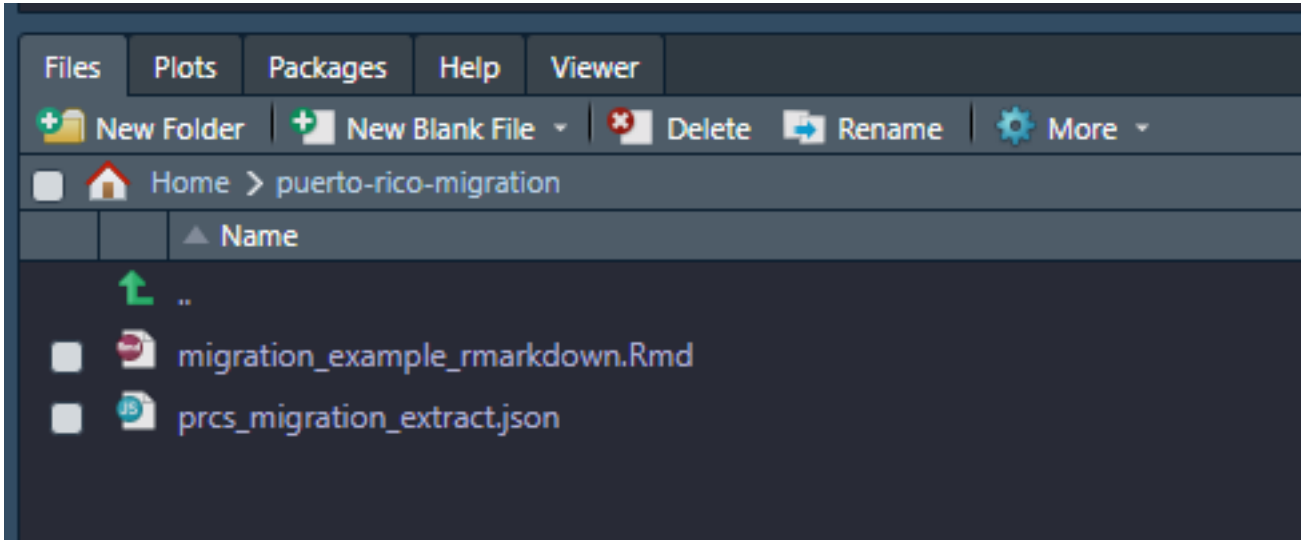
```
ddi <- read_ipums_ddi("prcs_migration_extract.xml")
data <- read_ipums_micro(
  ddi,
  data_file = "prcs_migration_extract.dat.gz"
)
```

Then hard-code the renamed files

Now this will work for anyone!

```
ddi <- read_ipums_ddi("prcs_migration_extract.xml")
data <- read_ipums_micro(
  ddi,
  data_file = "prcs_migration_extract.dat.gz"
)
```

Create a GitHub repository

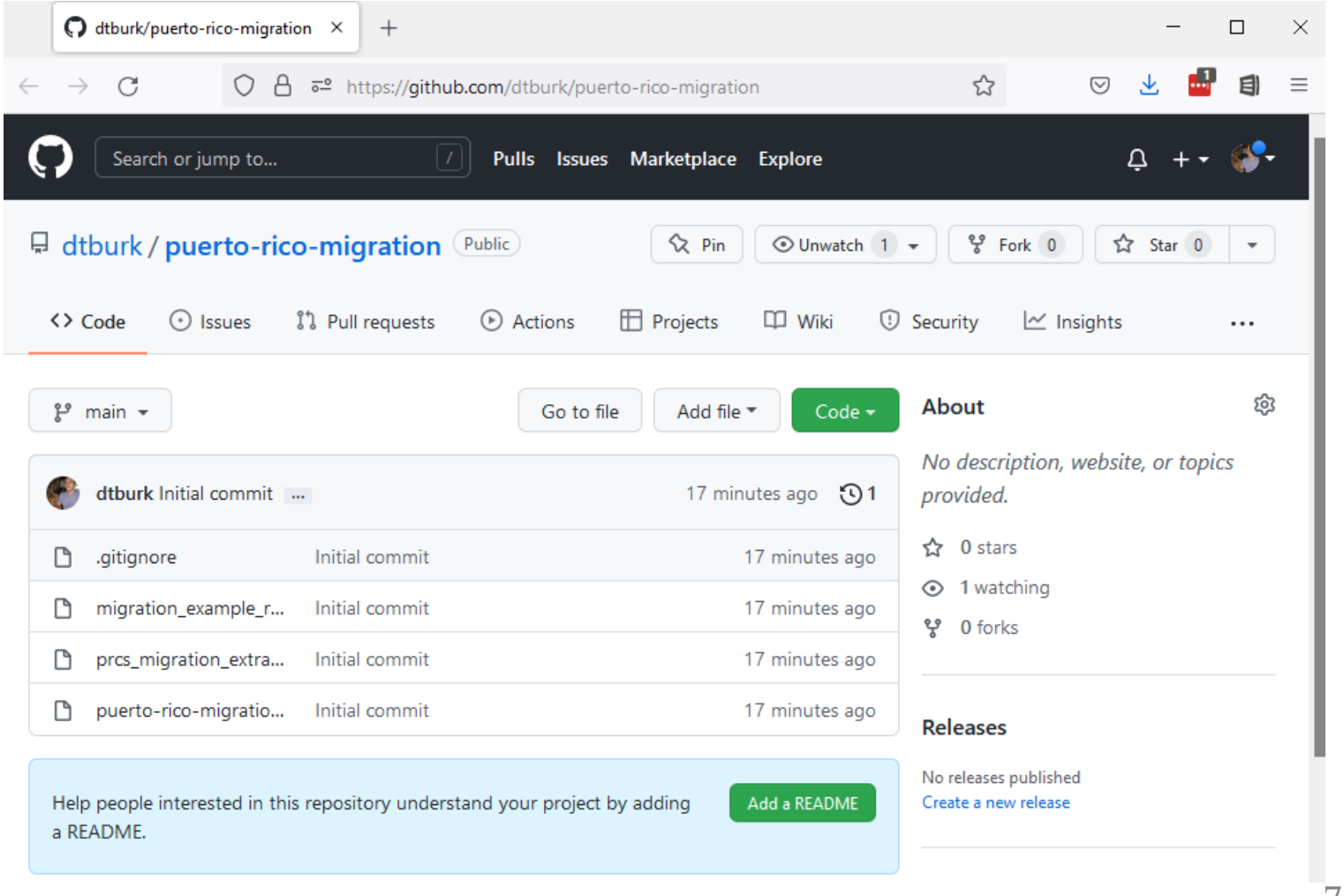


Create a GitHub repository

```
usethis::create_project(".")
usethis::use_git()
usethis::use_github()
```

For more details, check out the free online book Happy Git and GitHub for the useR

Create a GitHub repository



Resources

- Email us: ipums+cran@umn.edu
- Post on the IPUMS User Forum: https://forum.ipums.org/
- This presentation: https://github.com/ipums/ipumsr-psu-api-workshop
- ipumsr website, with vignettes: http://tech.popdata.org/ipumsr/index.html
- IPUMS USA API vignette: https://tech.popdata.org/ipumsr/dev/articles/ipums-api.html
- IPUMS tutorials page: https://www.ipums.org/support/tutorials
- IPUMS NHGIS API documentation: https://developer.ipums.org/docs/workflows/
- Instructions for using replicate weights: https://usa.ipums.org/usa/repwt.shtml
- *Happy Git and GitHub for the useR*: https://happygitwithr.com