IPUMS Data Training Exercise:

CPS Extraction and Analysis

(Exercise 1 for R)

Learning goals

- Gain an understanding of how the IPUMS dataset is structured and how it can be leveraged to explore your research interests.
- Create and download an IPUMS data extract
- Decompress data file and read data into R
- Analyze the data using sample code
- Validate data analysis work using answer key

Summary

This exercise will use the IPUMS dataset to explore associations between health and work status and to create basic frequencies of food stamp usage and answer the following research question: "What is the frequency of food stamp recipiency in the US? Are health and work statuses related?" You will create a data extract that includes the variables PERNUM, FOODSTMP, AGE, EMPSTAT, AHRSWORKT, and HEALTH; then you will use sample code to analyze these data.

R Code to Review

This tutorial's sample code and answers use the so-called "tidyverse" style, but R has the blessing (and curse) that there are many different ways to do almost everything. If you prefer another programming style, please feel free to use it. But, for your reference, these are some quick explanations for commands that this tutorial will use:

Code	Purpose
%>%	The pipe operator which helps make code with nested function calls is easier to read. When reading code, read as "and then". The pipe make it so that the code like **ingredient %>% stir() %>% cook()** is equivalent to cook (stir(ingredients)) (read as "take *ingredients* and then *stir* and then *cook*").
as_factor	Converts the value labels provide for IPUMS data into a factor variable for R
summarize	Summarize a datasets observations to one or more groups
group_by	Set the groups for the summarize function to group by
filter	Filter the dataset so that it only contains these variables
mutate	Add on a new variable to a dataset
ggplot	Make graphs using ggplot2
weighted.mean	Get the weighted mean of the "a" variable

Common Mistakes to Avoid

1. Not changing the working directory to the folder where your data is stored
2. Mixing up = and == ; To assign a value in generating a variable, use "<-" (or "="). Use "==" to test for equality.

Note*:* In this exercise, for simplicity we will use "weighted.mean". For analyses where variance estimates that take the survey design into consideration, use either the "survey" or "srvyr" package instead.

Registering with IPUMS

Go to http://cps.ipums.org, click on Register with IPUMS and apply for access. On login screen, enter email address and password and submit it!

Creating and downloading an IPUMS data extract

Make an Extract

- Go back to homepage and go to Select Data
- Click the Select Samples box, check the box for the 2011 ASEC sample, Click the Submit sample selections box

- Using the drop down menu or search feature, select the following variables:
  - FOODSTMP: Food stamp receipt
  - AGE: Age
  - EMPSTAT: Employment status
  - AHRSWORKT: Hours worked last week
  - HEALTH: Health status

Request the Data

- Click the VIEW CART button under your data cart
- Review variable selection. Click the Create Data Extract button
  - Review the 'Extract Request Summary' screen
  - Data format should be set to ".dat (fixed-width text)"
  - Describe your extract and click Submit Extract
- You will get an email when the data is available to download
  - To get to the page to download the data, follow the link in the email, or follow the Download and Revise Extracts link on the homepage

Download the Data

- Go to http://cps.ipums.org and click on Download or Revise Extracts
  - Right-click on the data link next to extract you created
  - Choose "Save Target As..." (or "Save Link As...")
  - Save into "Documents" (that should pop up as the default location)
- Do the same thing for the DDI link next to the extract
- (Optional) Do the same thing for the R command file
- You do not need to decompress the data to use it in R

Getting the data into your statistics software

The following instructions are for R. If you would like to use a different stats package, see: http://cps.ipums.org/cps/extract_instructions.shtml

Install the ipumsr package

- Open R from the Start menu
- If you haven't already installed the ipumsr package, in the command prompt, type the following command:

```
install.packages("ipumsr")
```

Read in the Data

- Set your working directory to where you saved the data above by adapting the following command (Rstudio users can also use the "Project" feature to set the working directory. In the menubar, select File -> New Project -> Existing Directory and then navigate to the folder):

```
setwd("~/")

# "~/" goes to your Documents directory on most computers
```

- Run the following command from the console, adapting it so it refers to the extract you just created (note the number may not be the same depending on how many extracts you've already made):

```
library(ipumsr)
```

```
ddi <- read_ipums_ddi("cps_00001.xml")

data <- read_ipums_micro(ddi)

# Or, if you downloaded the R script, the following is
equivalent:

#    source("cps_00001.R")
```

- This tutorial will also rely on the dplyr package, so if you want to run the same code, run the following command (but if you know other ways better, feel free to use them):

```
library(dplyr)
```

- To stay consistent with the exercises for other statistical packages, this exercise does not spend much time on the helpers to allow for translation of the way IPUMS uses labelled values to the way base R does. You can learn more about these in the value-labes vignette in the R package. From R run command: vignette("value-labels", package = "ipumsr")

Analyzing the Sample

Part I: Frequencies of FOODSTMP

1. On the website, find the codes page for the FOODSTMP variable and write down the code value, and what category each code represents.

   _____

2. What is the universe for FOODSTMP in 2011 (under the Universe tab on the website)?

   _____

3. How many households received food stamps in 2011?

   _____

```
data %>%
     group_by(MORTGAGE = haven::as_factor(MORTGAGE)) %>%
     summarize(n = n())
```

4. What proportion of households received food stamps in 2011?

   _____

Using household weights (ASECWTH)

Suppose you were interested not in the number of people living in homes that received food stamps, but in the number of households that were food stamp participants. To get this statistic you would need to use the household weight.

In order to use household weight, you should be careful to select only one person from each household to represent that household's characteristics. This is accomplished using the "filter(pernum==1)" qualifier in the example code below. You will need to apply the household weight (ASECWTH).

5. How many households received food stamps in 2011? _____
6. What proportion of households received food stamps in 2011?

   _____

```
data %>%

     filter(PERNUM == 1) %>%

     summarize(ON_FOODSTMP = weighted.mean(FOODSTMP == 2,

     ASECWTH))
```

Part II: Relationships in the Data

7. What is the universe for EMPSTAT in 2011? _____
8. What are the possible responses and codes for the self-reported HEALTH variable?

   _____

9. What percent of people with 'poor' self-reported health are at work?

   _____

```
data %>%

   group_by(HEALTH = as_factor(HEALTH)) %>%

   summarize(AT_WORK = weighted.mean(EMPSTAT == 10, ASECWT))
```

10. What percent of people with 'very good' self-reported health are at work?

    _____

11. In the EMPSTAT universe, what percent of people:
    a. self-report 'poor' health and are at work? _____
    b. self-report 'very good' health and are at work? _____

```
data %>%

     filter(AGE >= 15) %>%

     mutate(AT_WORK = EMPSTAT == 10) %>%

     group_by(HEALTH, AT_WORK) %>%

     summarize(n = sum(ASECWT)) %>%

     mutate(pct = n / sum(n))
```

Note: both AGE >= 15 and EMPSTAT != 0 will ensure that you are in the EMPSTAT universe.

Part III: Relationships in the Data

12. What is the universe for AHRSWORKT? _____
13. What are the average hours of work for each self-reported health category?

    _____

```
data %>%
     filter(AGE >= 15, AHRSWORKT < 999) %>%
     group_by(HEALTH = as_factor(HEALTH)) %>%
     summarize(AHRSWORKT = weighted.mean(AHRSWORKT, ASECWT))
```

Answers

Part I: Frequencies of FOODSTMP

1. On the website, find the codes page for the FOODSTMP variable and write down the code value, and what category each code represents. 0 NIU; 1 No; 2 Yes
2. What is the universe for FOODSTMP in 2011 (under the Universe tab on the website)? All interviewed households and group quarters. Note the NIU on the codes page, this is a household variable and the NIU cases are the vacant households.
3. How many people received food stamps in 2011? 39,187,407

```
data %>%

summarize(FOODSTMP = weighted.mean(FOODSTMP == 2, ASECWT))

#> # A tibble: 1 x 1

#>   FOODSTMP

#>          <dbl>

#> 1 0.1278321
```

4. What proportion of the population received food stamps in 2011? 12.78%

Using household weights (ASECWTH)

5. How many households received food stamps in 2011? 12,855,335 households
6. What proportion of households received food stamps in 2011? 10.71% of households

```
data %>%

     filter(PERNUM == 1) %>%

     summarize(ON_FOODSTMP = weighted.mean(FOODSTMP == 2,
     ASECWTH))

     #> # A tibble: 1 x 1

     #>   ON_FOODSTMP

#>         <dbl>

#> 1      0.1071324
```

Part II: Relationships in the Data

7. What is the universe for EMPSTAT in 2011? age 15+
8. What are the possible responses and codes for the self-reported HEALTH variable? Excellent; 2 Very Good; 3 Good; 4 Fair; 5 Poor
9. What percent of people with 'poor' self-reported health are at work? 11.61%

```
data %>%

  group_by(HEALTH = as_factor(HEALTH)) %>%

  summarize(AT_WORK = weighted.mean(EMPSTAT == 10, ASECWT))

#> # A tibble: 5 x 2

#>   HEALTH AT_WORK

#>   <fctr> <dbl>

#> 1 Excellent 0.4332012

#> 2 Very good 0.5161936

#> 3      Good 0.4535441

#> 4      Fair 0.2707342

#> 5      Poor 0.1160629
```

10. What percent of people with 'very good' self-reported health are at work? 51.62%

11. In the EMPSTAT universe, what percent of people:
    a. self-report 'poor' health and are at work? 11.81%
    b. self-report 'very good' health and are at work? 63.9%

```
data %>%
     filter(AGE >= 15) %>%
     mutate(AT_WORK = EMPSTAT == 10) %>%
     group_by(HEALTH, AT_WORK) %>%
     summarize(n = sum(ASECWT)) %>%
     mutate(pct = n / sum(n))
#> # A tibble: 10 x 4
#> # Groups: HEALTH [5]
#>        HEALTH AT_WORK      n    pct
#>   <int+lbl> <lgl> <dbl> <dbl>
#> 1       1 FALSE 23974866 0.3609944
#> 2       1 TRUE 42438528 0.6390056
#> 3       2 FALSE 28004656 0.3601949
#> 4       2 TRUE 49743963 0.6398051
#> 5       3 FALSE 31684784 0.4793809
#> 6       3 TRUE 34410426 0.5206191
#> 7       4 FALSE 17666680 0.7176749
#> 8       4 TRUE 6949870 0.2823251
#> 9       5 FALSE 9259361 0.8819270
#> 10      5 TRUE 1239650 0.1180730
```

Part III: Relationships in the Data

12. What is the universe for AHRSWORKT? Civilians age 15+, at work last week

13. What are the average hours of work (even if they did not work last week) for each self-reported health category? Excellent – 39.40; Very Good – 38.66; Good – 37.78; Fair – 34.67; Poor – 32.41

```
data %>%
     filter(AGE >= 15, AHRSWORKT < 999) %>%
     group_by(HEALTH = as_factor(HEALTH)) %>%
     summarize(AHRSWORKT = weighted.mean(AHRSWORKT, ASECWT))
#> # A tibble: 5 x 2
#>   HEALTH AHRSWORKT
#>   <fctr> <dbl>
#> 1 Excellent 38.39723
#> 2 Very good 38.66232
#> 3      Good 37.77809
#> 4      Fair 35.67317
#> 5      Poor 32.41218
```