Assignment 1b: Replicating *Sexual Orientation and Labor Economics* by Cara Brown (1998)

Daniel Sánchez

1/12/23

This document walks through the replication of Cara Brown's 1998 paper *Sexual Orientation and Labor Economics* published at *Feminist Economics*.

Preliminaries

I need the following R libraries to make this code work:

```
library(tidyverse) # For data wrangling
library(haven) # Reading stata files
library(labelled) # For survey labels
library(modelsummary) # For tables
library(data.table) # For combinations
```

I load the data from the given file below:

```
df<-
  read_dta('data/Census_2016_Hierarchial.dta')
```

Data Cleaning

In this subsection I do some data cleaning and exploratory analysis so that I build a fit dataframe for the replication study.

Below, I clean the income and age group variables to have them as I'd need to in the replication. Note I use *haven*'s `as_factor()`, not base R `as.factor()`. We also filter for the ages we'd

like to use, then create another age group which combines 45 to 49 with 50 to 54 to create 45-54.

```
# First, rename the income variable (empin) to something more usable
# Also, we apply value labels as necessary in the mutate() call

df<-
  df %>%
  rename(income = 'empin') %>%
  mutate(agegrp = case_when(
    agegrp ==  88 ~ as.double(NA),
    TRUE ~ as.double(agegrp)
  ))

# Now, replace the income invalid values (88,888,888, 99,999,999) with actual NAs in R.

df <-
  df %>%
  mutate(income_clean = case_when(
      income %in% c(88888888, 99999999) ~ as.numeric(NA),
      TRUE ~ income
    )
  )

df %>% select(PP_ID, agegrp, income_clean) %>% head()
```

```
# A tibble: 6 x 3
  PP_ID agegrp income_clean
  <dbl>  <dbl>        <dbl>
1 11101     10        64000
2 21101      6        47000
3 21102      7           NA
4 21103      1           NA
5 31101     10        43000
6 41101      8        58000
```

```
# Do the same data cleaning with gender

df <-
  df %>%
  mutate(sex = case_when(
```

```
    sex == 8 ~ as.double(NA),
    TRUE ~ as.double(sex)
    ))
```

Let us group by person and count how much records I have for that. Test to see if there is more than one observation per person

```
people <-
  df %>%
  group_by(PP_ID) %>%
  summarise(obs= n())

people %>% filter(obs != 1)
```

```
# A tibble: 0 x 2
# ... with 2 variables: PP_ID <dbl>, obs <int>
```

There is only one observation per person in the dataset, as expected

Constructing the presumed homosexual variable

I now construct a dummy variable `hmsxl` which is 1 if the individual is a presumed homosexual and 0 otherwise. The variable is complicated to build as it depends on many things.

First, I will pay attention to the people who live with non-relatives only. I can see this variable from the `cfstat` variable in the census. These are potential couples to someone else, so I will create a dummy for these people, `non_rel`.

```
# Create the variable

df <-
  df %>%
  mutate(non_rel = ifelse(cfstat == 7, 1, 0))

# Now, do the grouping at the household level, filtering out the people whose age we're no

households <-
  df %>%
  filter(agegrp %in% c(9,10,11)) %>%
  group_by(HH_ID) %>%
```

```
  summarise(non_rel = sum(non_rel),
            people = n(),
            sexes = n_distinct(sex)) %>%
  arrange(desc(non_rel))

head(households)
```

```
# A tibble: 6 x 4
   HH_ID non_rel people sexes
   <dbl>   <dbl>  <int> <int>
1  98015       7      7     1
2   5948       4      4     2
3 102220       4      4     1
4 126135       4      4     1
5  17901       3      3     1
6  20045       3      5     1
```

We will determine if a household potentially has homosexual people in it (individuals unrelated through family) by marking households which have a greater number of people in the house compared to the number of genders. If there is, for instance, a house with 3 people in it where at least one of them is unrelated in family terms and only two genders, there is a potential homosexual couple living in the household. As I have no more than this, I have no choice to later "mark" all of the people living in the household as homosexual.

Below, I will create the `hmsxl` indicator at the household level, and later left join this data to the individual level dataframe to mark each individual as presumed homosexuals.

```
# Create the indicator

households <-
  households %>%
  mutate(hmsxl = ifelse(non_rel >= 1 & people > sexes, 'Homosexual','Heterosexual')) %>%
  select(HH_ID, hmsxl)

# Make it at the individual level

df_joint <-
  df %>%
  filter(agegrp %in% c(9,10,11)) %>%
  left_join(households, by = 'HH_ID') %>%
  mutate(hmsxl = as.factor(hmsxl))
```

I can now compute conditional means for income based on ages and whether or not they're homosexual.

```
df_table<-
  df_joint %>%
  mutate(agegrp = case_when(
         agegrp %in% c(9,10) ~ '45-54',
         agegrp == 11 ~ '55-64'),
         sex = ifelse(sex == 1, 'F', 'M'),
         hmsxl = as.factor(hmsxl),
         MarStH = as_factor(MarStH)) %>%
  rename(Income = 'income_clean',
         Orientation = 'hmsxl',
         MarStatus = 'MarStH') %>%
  select(Orientation, agegrp, Income, sex, MarStatus)

datasummary((Income * Orientation* MarStatus) ~ (Mean*sex*agegrp),
            data = df_table,
            output = 'markdown')
```

			F / 45-54	F / 55-64	M / 45-54	M / 55-64
Income	Heterosexual	Never legally married (and not living common law)	50842.81	45897.56	52505.92	42662.13
		Legally married (and not separated)	51425.14	41285.84	86195.94	71999.94
		Living common law	47545.67	40385.89	67383.56	59784.23
		Separated, divorced or widowed (and not living common law)	48365.14	40371.26	66339.22	54662.86
		Not available	60500.00	51000.00	63666.67	
	Homosexual	Never legally married (and not living common law)	46604.59	37851.87	39171.43	36387.53
		Legally married (and not separated)	41000.00	26300.05	50461.54	38515.15
		Living common law	70078.88	42500.17	132443.00	35750.00
		Separated, divorced or widowed (and not living common law)	34829.81	38421.88	43468.09	35149.25
		Not available				

Doing the same with a pairwise approach

An alternative method is to start with the database from scratch as it is, filter out for the people that we want, and then define a new data frame that now creates all possible pairs of two people. We will filter out the dataframe for the people that we need, then

```
df_for_pairs <-
  df %>%
  filter(agegrp %in% c(9,10,11),
         MarStH == 1) %>%
  mutate(person_id = as.character(PP_ID))

pairs <-
  CJ(df_for_pairs$PP_ID, df_for_pairs$PP_ID, unique = T) %>%
  rename(person1 = 'V1', person2 = 'V2') %>%
  mutate(identical = (person1 == person2)) %>%
  filter(identical == FALSE)

# combinations <- combn(unique(df_for_pairs$PP_ID),2) %>% t() This takes too long
```

Having done the pairs, we will need to get their household information as well as their sex and other variables as well. We use joins for this.