Constructing the MPI

Nate Kratzer

April 9, 2017

The Overall Structure

A multidimensional poverty index attempts to capture a more accurate measurement of how poverty is actually experienced. Traditionally, poverty has been measured solely by income. While this may be a decent proxy in some cases, it does mean giving zero weight to other important parts of life. Multidimeensional poverty indexes have become widely used in international development. The Oxford Poverty and Human Development Initiative sums up the reasons for switching from income-based to multidimensional measures of poverty: "empirically, income-poor households are (surprisingly) not well-matched to household carrying other basic deprivations like malnutrition; also the trends of income and non-income deprivations are not matched, and nor does growth ensure the reduction of social deprivations. And, a dashboard overlooks the interconnection between deprivations, which people experience and policies seek to address." (Alkire et al 2015)

This document shows how to construct a multidimensional poverty index for the United States using Census data from the American Community Survey and code in the R programming language.

The index uses nine indicators across six dimensions

1. Economic -Income Poverty -Employment
2. Health -Insurance -Disability
3. Housing -Cost of Housing -Overcrowding
4. Social Connections -Linguistic Isolation -Internet Access
5. Education -Educational Attainment

The initial data

Data comes from IPUMS USA and covers 2013, 2014, and 2015. I have commented out the lines of code used to read in the data from the original downloaded csv. After loading, I suggest saving in R's feather format to save time.

```r
library(dplyr)
library(readr)
library(feather)

#data = read_csv("C:/Users/natek/Documents/usa_00007.csv")

#write_feather(data, "C:/Users/natek/Documents/MPI feather.feather")

data = read_feather("C:/Users/natek/Documents/MPI feather.feather")

str(data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    9412410 obs. of  50 variables:
##  $ YEAR      : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ DATANUM   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ SERIAL    : int  1 2 2 2 2 3 4 4 4 5 ...
##  $ HHWT      : int  65 51 51 51 51 449 16 16 16 52 ...
##  $ STATEFIP  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTYFIPS: int  3 0 0 0 0 0 0 0 0 0 ...
##  $ METRO     : int  4 0 0 0 0 0 1 1 1 0 ...
##  $ MET2013   : int  19300 0 0 0 0 0 0 0 0 0 ...
##  $ CITY      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PUMA      : int  2600 2500 2500 2500 2500 1700 2200 2200 2200 2400 ...
##  $ GQ        : int  4 1 1 1 1 1 1 1 1 1 ...
##  $ FARM      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ OWNCOST   : int  99999 426 426 426 426 926 99999 99999 99999 99999 ...
##  $ RENTGRS   : int  0 0 0 0 0 0 370 370 370 140 ...
##  $ HHINCOME  : int  9999999 151000 151000 151000 151000 39930 11400 11400 11400 3900 ...
##  $ FOODSTMP  : int  1 1 1 1 1 1 2 2 2 2 ...
##  $ LINGISOL  : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ BEDROOMS  : int  0 4 4 4 4 4 5 5 5 2 ...
##  $ CILAPTOP  : int  0 1 1 1 1 1 2 2 2 2 ...
##  $ CIHAND    : int  0 2 2 2 2 2 1 1 1 2 ...
##  $ CIOTHCOMP : int  0 2 2 2 2 2 2 2 2 2 ...
##  $ CINETHH   : int  0 1 1 1 1 3 3 3 3 3 ...
##  $ PERNUM    : int  1 1 2 3 4 1 1 2 3 1 ...
##  $ PERWT     : int  65 51 62 232 97 449 16 30 7 52 ...
##  $ FAMSIZE   : int  1 4 4 4 4 1 3 3 3 1 ...
##  $ FAMUNIT   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ SEX       : int  2 2 1 1 2 2 2 1 1 2 ...
##  $ AGE       : int  19 55 56 21 21 63 61 20 12 78 ...
##  $ RACE      : int  1 2 2 2 2 2 3 3 3 2 ...
##  $ RACED     : int  100 200 200 200 200 200 308 308 308 200 ...
##  $ HISPAN    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HISPAND   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CITIZEN   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HCOVANY   : int  2 2 2 2 2 2 2 1 2 2 ...
##  $ EDUC      : int  7 8 6 7 7 10 5 6 2 0 ...
##  $ EDUCD     : int  71 81 63 71 71 101 50 63 23 2 ...
##  $ GRADEATT  : int  6 0 0 6 6 0 0 0 4 0 ...
##  $ GRADEATTD : int  60 0 0 60 60 0 0 0 43 0 ...
##  $ EMPSTAT   : int  3 1 1 3 3 1 3 2 0 3 ...
##  $ EMPSTATD  : int  30 10 10 30 30 10 30 20 0 30 ...
##  $ WKSWORK2  : int  0 6 6 0 0 6 0 1 0 0 ...
##  $ INCTOT    : int  0 52000 99000 0 0 39930 10300 1100 9999999 3900 ...
##  $ FTOTINC   : int  9999999 151000 151000 151000 151000 39930 11400 11400 11400 3900 ...
##  $ INCWAGE   : int  0 52000 0 0 0 39000 0 1100 999999 0 ...
##  $ INCINVST  : int  0 0 0 0 0 0 0 0 999999 0 ...
##  $ INCRETIR  : int  0 0 0 0 0 0 0 0 999999 0 ...
##  $ POVERTY   : int  0 501 501 501 501 330 61 61 61 35 ...
##  $ DIFFMOB   : int  1 1 1 1 2 1 2 1 0 1 ...
##  $ DIFFCARE  : int  1 1 1 1 2 1 1 1 1 1 ...
##  $ TRANTIME  : int  0 30 0 0 0 15 0 0 0 0 ...
```

Economic

Income Poverty

The income poverty indicator is straightforward and is the same for all members of the household. The ACS data already contains a poverty indicator showing percent of poverty line for all households. I recode this as a binary variable such that households are poor (1) or not poor (0)

```
data$income_poverty = if_else(data$POVERTY < 100, 1, 0)
```

Family Employment

Family employment is also assigned at the household level. If no one in the family is employed, then the family is initially considered employment deprived. However, if anyone in the family has retirement income, or investment income in excess of $25,000, or if the family lives on a working farm, then the family is not considered employment deprived.

```
data = data %>%
  mutate(emp_num = if_else(EMPSTAT == 1, 1, 0)) %>%
  group_by(SERIAL, YEAR) %>%
  mutate(retired = if_else(any(INCRETIR > 1), 1, 0),
         investment = if_else(any(INCINVST > 25000), 1, 0),
         fam_emp = if_else(any(emp_num ==1)|any(retired ==1)|any(investment == 1)|FARM == 2, 0, 1))
```

Health

Health Insuranc

Health insurance is assigned at the individual level. Some members of the household may have health insurance, while others do not.

```
data$health = if_else(data$HCOVANY == 2, 0, 1)
```

Disability

Disability, for this index, is defined as those who said either they had difficulty caring for themselves or independent living difficulty.

```
data$disability <- if_else(data$DIFFMOB == 2 |data$DIFFCARE == 2, 1, 0)
```

Education

Education deprivation is at the individual level. It is calculated differently for those under the age of 19. For those 19 and older, a lack of high school degree is considered educationally deprived. For those under 18, being in a grade lower than one's age warrants is considered to be educationally deprived. For example, a 7-year-old who is attending kindergarten or an 18-year-old who is attending 11th grade.

```
data$education = if_else(data$EDUCD < 62 & data$AGE > 18 & data$EDUCD !=1, 1, 0) #less than hs, over 18

data$grade_recode = NA ##where GRADEATTD == 0, it stays NA, everywhere else it changes. 0 is NA in GRADE
data$grade_recode[data$GRADEATTD < 30 & data$GRADEATTD != 0] = 0
data$grade_recode[data$GRADEATTD == 30] = 4
data$grade_recode[data$GRADEATTD == 31] = 1
data$grade_recode[data$GRADEATTD == 32] = 2
data$grade_recode[data$GRADEATTD == 33] = 3
data$grade_recode[data$GRADEATTD == 34] = 4
data$grade_recode[data$GRADEATTD == 40] = 8
data$grade_recode[data$GRADEATTD == 41] = 5
data$grade_recode[data$GRADEATTD == 42] = 6
data$grade_recode[data$GRADEATTD == 43] = 7
data$grade_recode[data$GRADEATTD == 44] = 8
```

```
data$grade_recode[data$GRADEATTD == 50] = 12
data$grade_recode[data$GRADEATTD == 51] = 9
data$grade_recode[data$GRADEATTD == 52] = 10
data$grade_recode[data$GRADEATTD == 53] = 11
data$grade_recode[data$GRADEATTD == 54] = 12
data$grade_recode[data$GRADEATTD > 59] = 120


data$education[data$AGE < 19 & ((data$GRADEATTD+6) < data$AGE)] = 1 ##If age is more than 6 years above
```

Housing Quality

Overcrowding

Overcrowding is assigned at the household level. The individuals in the household are assigned the status of overcrowded if there are more than two people per bedroom.

```
data = data %>%
  group_by(SERIAL, YEAR) %>%
  mutate(hhsize = n())


data$overcrowd = if_else(data$hhsize > 2*data$BEDROOMS, 1, 0)
```

Housing costs

Houses are considered cost burdened at above 30% of total income and severely cost burdened at over 50% of total income. Here I follow Dhongde and Haveman 2016 in using severe cost burden as the indicator.

```
data$OWNCOST[data$OWNCOST == 99999] <- NA
data$FTOTINC[data$FTOTINC == 99999] <- NA

data$hcost <- apply(data[ ,c("OWNCOST", "RENTGRS")], 1, max, na.rm = TRUE)
data$hcost_dep <- if_else(data$hcost*12/data$FTOTINC > .50, 1, 0) # HUD uses 50% as severe cost burdene
```

Social Connections

Internet/Computer Access

Internet access is assigned at the household level. A household is considered deprived if there is either no internet access, or there is none of the following computer equipment: laptop, desktop, notebook, smartphone, other computer equipment (not including GPS or household appliances).

```
data$computer_internet <- if_else(data$CINETHH == 3|(data$CILAPTOP == 2 & data$CIHAND == 2 & data$CIOTH
```

Linguistic isolation

Linguistic isolation is calculated as not having anyone in the household over the age of 14 who speaks English. It is calculated by the census bureau.

```
data$lang_dep <- if_else(data$LINGISOL == 2, 1, 0)
```

Subsetting the Data

Saving only the data that is used to construct the index, as well as a few groups it can be broken into (e.g. by state, gender, etc.)

```
mpi_data = data %>%
  select(YEAR, SERIAL, PERWT, SEX, METRO, STATEFIP, income_poverty, fam_emp, health, education, overcrow
```

Classifying Individuals as MPI poor

Each of the five dimensions is equally weighted (.2). For the dimensions with two indicators they are equally weighted (.1) while the Education indicator has the full weight of the education dimension (.2).

I define MPI poor as being poor in 3 or more weighted indicators, with Education being weighted at .2 because it is the only indicator in its dimension. Thus, a weighted score of over .29 is MPI poor.

```
mpi_data$mpi_score = mpi_data$income_poverty*.1 +
  mpi_data$fam_emp *.1 +
  mpi_data$health*.1 +
  mpi_data$education*.2 +
  mpi_data$overcrowd*.1 +
  mpi_data$computer_internet*.1 +
  mpi_data$disability*.1 +
  mpi_data$lang_dep * .1 +
  mpi_data$hcost_dep * .1

mpi_data$mpi_poor = if_else(mpi_data$mpi_score > .29, 1, 0)

mpi_data_15 <- mpi_data %>% filter(YEAR == 2015)
```

Writing out the data.

```
write_feather(mpi_data, "C:/Users/natek/Documents/mpi_data.feather")
write_feather(mpi_data_15, "C:/Users/natek/Documents/mpi_data_15.feather")
```

National Alkire-Foster Index

Here I calculate an AF index for the U.S. in 2015.

```
headcount = sum(mpi_data_15$mpi_poor*mpi_data_15$PERWT, na.rm = TRUE)/sum(mpi_data_15$PERWT)

income_censored_hc = sum(mpi_data_15$income_poverty[mpi_data_15$mpi_poor == 1]* mpi_data_15$PERWT[mpi_da

fam_emp_censored_hc = sum(mpi_data_15$fam_emp[mpi_data_15$mpi_poor == 1]* mpi_data_15$PERWT[mpi_data_15

health_censored_hc = sum(mpi_data_15$health[mpi_data_15$mpi_poor == 1]* mpi_data_15$PERWT[mpi_data_15$mp

education_censored_hc = sum(mpi_data_15$education[mpi_data_15$mpi_poor == 1]* mpi_data_15$PERWT[mpi_dat

overcrowd_censored_hc = sum(mpi_data_15$overcrowd[mpi_data_15$mpi_poor == 1]* mpi_data_15$PERWT[mpi_dat

computer_internet_censored_hc = sum(mpi_data_15$computer_internet[mpi_data_15$mpi_poor == 1]* mpi_data_

disability_censored_hc = sum(mpi_data_15$disability[mpi_data_15$mpi_poor == 1]* mpi_data_15$PERWT[mpi_da
lang_dep_censored_hc = sum(mpi_data_15$lang_dep[mpi_data_15$mpi_poor == 1]*mpi_data_15$PERWT[mpi_data_15

hcost_dep_censored_hc = sum(mpi_data_15$hcost_dep[mpi_data_15$mpi_poor == 1]*mpi_data_15$PERWT[mpi_data_

mpi_poor_hc = sum(mpi_data_15$mpi_poor*mpi_data_15$PERWT, na.rm = TRUE)
n = sum(mpi_data_15$PERWT)
income_per = sum(mpi_data_15$income_poverty * mpi_data_15$PERWT)/n
fam_emp_per = sum(mpi_data_15$fam_emp * mpi_data_15$PERWT)/n
health_per = sum(mpi_data_15$health * mpi_data_15$PERWT)/n
education_per = sum(mpi_data_15$education * mpi_data_15$PERWT)/n
overcrowd_per = sum(mpi_data_15$overcrowd * mpi_data_15$PERWT)/n
computer_internet_per = sum(mpi_data_15$computer_internet * mpi_data_15$PERWT)/n
disability_per = sum(mpi_data_15$disability * mpi_data_15$PERWT)/n
lang_dep_per = sum(mpi_data_15$lang_dep * mpi_data_15$PERWT)/n
hcost_dep_per = sum(mpi_data_15$hcost_dep * mpi_data_15$PERWT, na.rm =TRUE)/n


intensity = (income_censored_hc*.1 + fam_emp_censored_hc *.1 +
           health_censored_hc * .1 + education_censored_hc * .2 +
           overcrowd_censored_hc * .1 + computer_internet_censored_hc * .1 + disability_censored_hc *.1
           lang_dep_censored_hc*.1 + hcost_dep_censored_hc*.1)/ mpi_poor_hc
adj_hc = headcount*intensity
per_contrib_income = ((income_censored_hc/n)/adj_hc)*.1
per_contrib_fam_emp = ((fam_emp_censored_hc/n)/adj_hc)*.1
per_contrib_health = ((health_censored_hc/n)/adj_hc)*.1
per_contrib_education = ((education_censored_hc/n)/adj_hc)*.2
per_contrib_overcrowd = ((overcrowd_censored_hc/n)/adj_hc)*.1
per_contrib_disability = ((disability_censored_hc/n)/adj_hc)*.1
per_contrib_lang_dep = ((lang_dep_censored_hc/n)/adj_hc)*.1
per_contrib_hcost_dep = ((hcost_dep_censored_hc/n)/adj_hc)*.1
per_contrib_computer_internet = ((computer_internet_censored_hc/n)/adj_hc)*.1
```

Printing Results

Below I show the headcount, intensity, and adjusted headcount for the U.S. in 2015. I also show the percent of the populatoin experiencing each indicator and the percent contribution each indicator makes to the index.

```
headcount
```

```
## [1] 0.1539587
```

```
intensity
```

```
## [1] 0.3812201
```

```
adj_hc
```

```
## [1] 0.05869217
```

```
col1 <- c(income_per,fam_emp_per,health_per,education_per, overcrowd_per, computer_internet_per, disabil
        lang_dep_per, hcost_dep_per)
col1 <- round(col1, 2)
col2 <- c(per_contrib_income, per_contrib_fam_emp, per_contrib_health, per_contrib_education,
          per_contrib_overcrowd, per_contrib_computer_internet, per_contrib_disability, per_contrib_lan
          per_contrib_hcost_dep)
```

```r
col2 <- round(col2, 2)

ind_table <- cbind(col1, col2)
rownames(ind_table) <- c("Income", "Employment", "Health", "Education", "Overcrowding", "Internet", "Di
                         "Language", "Housing Cost")
colnames(ind_table) <- c("Incidence", "Contribution")

print(ind_table)
```

```
##              Incidence Contribution
## Income            0.17         0.17
## Employment        0.09         0.10
## Health            0.10         0.08
## Education         0.10         0.25
## Overcrowding      0.04         0.05
## Internet          0.14         0.12
## Disability        0.06         0.05
## Language          0.05         0.05
## Housing Cost      0.18         0.13
```