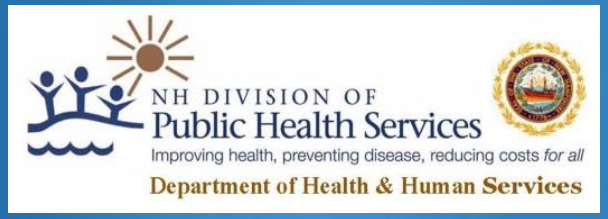
Using the R Language and `acs` Package to Compile and Update a Social Vulnerability Index for New Hampshire

Michael Laviolette, PhD MPH
Dennis Holt, MPH
Kristin K. Snow, ScD

State of New Hampshire
Department of Health and Human Services

American Community Survey Data Users' Conference
May 12, 2015



Background

- ACS replaces decennial Census “long form” with smaller ongoing samples
- Result: “warmer” (more current) but “fuzzier” (less precise)
- Census reports margin of error with all ACS data, but most users don’t know how or have tools to work with them

Nature of estimates

- "Estimate" is actually a point in a probability distribution; thus a degree of uncertainty accompanies any single number
- Practical applications often require combining estimates
- Determining standard error for combined estimates can be quite complex
  - E.g. when dividing estimates numerator can be subset of denominator (proportion) or not (ratio); different methods apply

Challenges of estimates

- Adjusting for inflation in multiyear estimates
- Overlapping errors
  - E.g. comparing 2006-2010 to 2007-2011
- Census reports 90% margin of error instead of familiar 95% confidence interval
- Aggregating non-count data (medians, means, percentages)
  - Different data types must be dealt with differently
  - New estimates must be built up from underlying counts
- Tedious, repetitive tasks are best automated

“R” environment

- R is a free, open-source environment for state-of-the-art statistical analysis and high-quality graphics
- Evolved from “S” language developed by AT&T Bell Labs in late 1970’s
- Runs under Windows, Mac OS, and Linux
- R project leadership includes more than 20 leading statisticians and computer scientists
- R users have created more than 6,600 add-on "packages" to expand functionality
- Popularity rapidly increasing in recent years

R data structures (objects)

- SAS, SPSS, Stata are “procedure-oriented” languages; R is “object-oriented”
- R has multiple data structures (“classes”)
  - e.g. data frame, matrix, lm (linear model fit)
  - Other languages have only one structure, data set
- R uses functions to create and manipulate objects; e.g. **`mean(x)`** gives mean of values in object **`x`**
- R does not produce voluminous output; rather it produces an object from which desired information is extracted (e.g. linear regression fit)
- New object types can be created by package developers

`acs` package

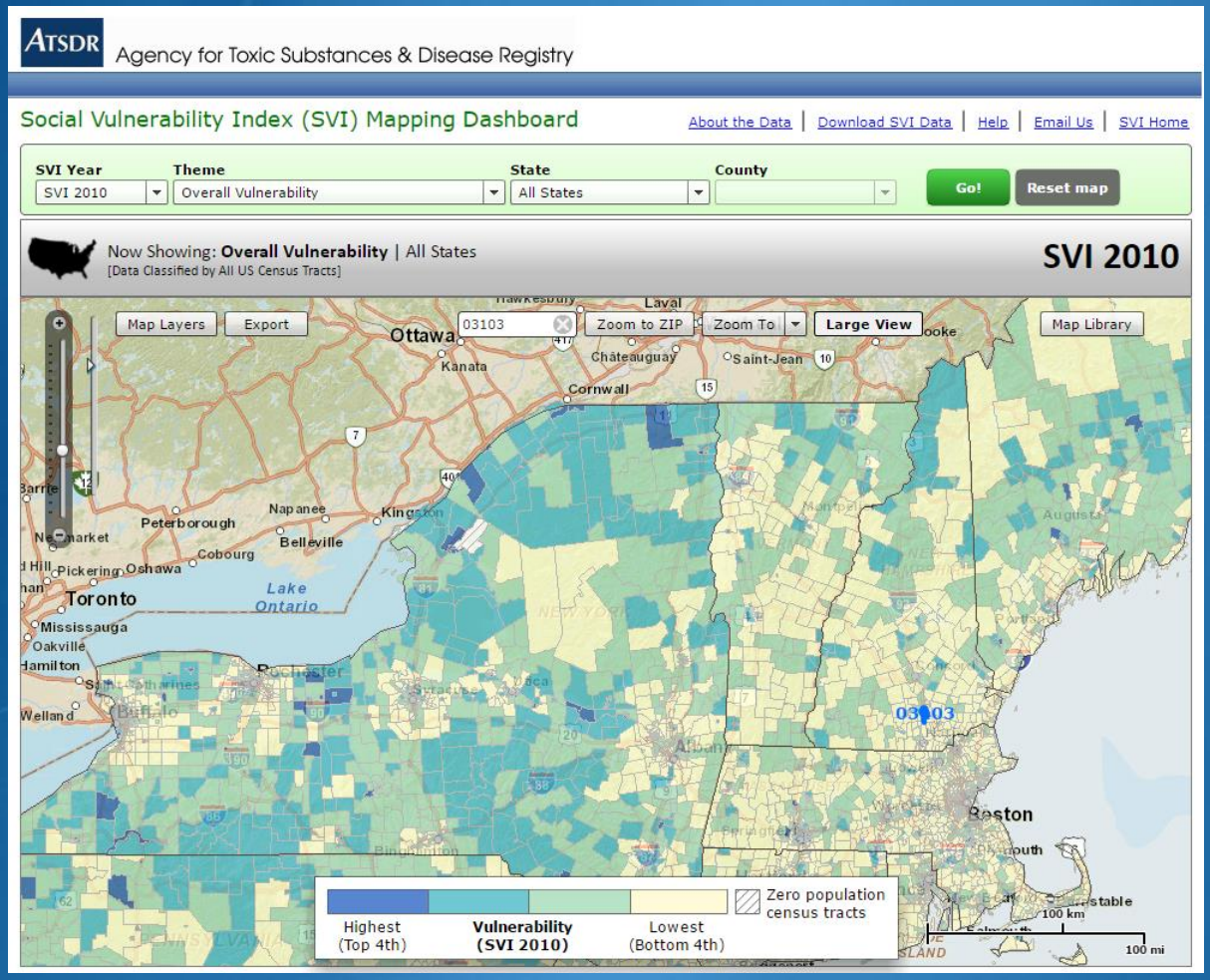
- Developed by Ezra Haber Glenn under contract with Puget Sound Regional Council
- Key features:
  - **`geo.set`** class to hold standard or user-defined geographies
  - **`acs`** class to hold ACS data
  - **`acs.fetch()`** function to download summary table data from Census API directly into workspace
  - Methods and convenience functions for proper handling of above objects; e.g. combining estimates and computing appropriate standard errors

Social vulnerability index (SVI)

- Social vulnerability refers to the resilience of communities when confronted by external stresses on human health, such as natural disasters or disease outbreaks
- SVI published by Flanagan et al. as CDC/ASTDR project
- Original SVI used 15 component measures from 2000 census
- We constructed SVI using American Community Survey (ACS) 2008-12 five-year summary estimates

Flanagan BE et al., "A Social Vulnerability Index for Disaster Management," *Journal of Homeland Security and Emergency Management* 8(1), ISSN (Online) 1547-7355, DOI: 10.2202/1547-7355.1792, January 2011.

National SVI at CDC



http://svi.cdc.gov

New Hampshire SVI Measures

Socioeconomic	1	Poverty, living below Federal poverty level
	2	Unemployed, age 16 and older and seeking work
	3	Per capita income (in 2013 inflation-adjusted $)
	4	Education, age 25+ without a high school diploma
	5	Health insurance, age less than 65 without insurance
Demographic	6	Children, population age younger than 18
	7	Elderly, population age 65 and older
	8	Disability, age 5 or older with a disability
	9	Single parent, households with children
	10	Minority, Hispanic or non-white race
	11	Limited English, age 5 and over who speak English less than "Well"
Housing and Transportation	12	Large apartment buildings, housing units 10 or more per building
	13	Mobile homes
	14	Crowding (housing units with more than one person per room)
	15	No vehicle (households with no vehicle available)
	16	Group quarters

All measures except per capita income are percentages of appropriate population universe

Original methods

- Downloaded CSV summary files from Census FTP site
- Identified needed tables from sequence number tables
- Read files into Microsoft Excel, merging with geography table
- Tedious, time-consuming, and error-prone
- Uncertainty not included in analysis
- R **`acs`** package presented opportunity for improvement

Setting up geography with `acs`

- Once geography is defined, multiple measures can be derived
- New Hampshire has 10 counties and 295 tracts
- Most straightforward geography is all tracts for each county using * wildcard
- Three tracts have zero population
  - Include Manchester airport and Atlantic Ocean
  - Zero-population tracts caused problems in computing percentages due to zero denominators
  - Geography had to be modified to list tracts explicitly for counties with zero-population tracts

Setting up geography (2)

```
install.packages("acs") # only need to do once
library(acs)
api.key.install("YOUR_KEY_GOES_HERE") # only need to do once

# Get all tracts in state in two steps
# Get list of all counties in state
nh.cty.lst <- acs.fetch(geography = geo.make(state = 33,
                                             county = "*"),
                        table.number = "B01003")
# Table B01003 is Total Population
# For each county, get all the tracts
nh.tract <-
  geo.make(state = 33,
           county = as.numeric(geography(nh.cty.lst)[[3]]),
           tract = "*")

class(nh.tract) # class "geo.set"
```

Setting up geography (3)

```
library(acs)
# Get list of counties
nh.cty.lst <- acs.fetch(geography = geo.make(state = 33, county = "*"),
                        table.number = "B01003")

# All tracts for counties except Hillsborough and Rockingham
geo.step1 <- geo.make(state = 33,
                      county = as.numeric(geography(nh.cty.lst)[[3]][-c(6,8)]),
                      tract = "*")

# Tracts for Hillsborough County, excluding tract 9801.11 (airport)
hil <- acs.fetch(geography = geo.make(state = 33, county = as.numeric(11),
                                      tract = "*"),
                 table.number = "B01003")
geo.step2 <-  geo.make(state = 33, county = as.numeric(11),
                       tract = as.numeric(geography(hil)[[4]][-86]))

# Tracts for Rockingham County, excluding 9800.11 (airport) and 9900 (ocean)
roc <- acs.fetch(geography = geo.make(state = 33, county = as.numeric(15),
                                      tract = "*"),
                 table.number = "B01003")
geo.step3 <-  geo.make(state = 33, county = as.numeric(15),
                       tract = as.numeric(geography(roc)[[4]][-(65:66)]))

# Combine geography objects, adding geography for state as a whole
nh.tract <- geo.step1 + geo.step2 + geo.step3 + geo.make(state = 33)

# working geography has 158 elements
```

Computing SVI measures

- Almost all measures are the percentage of the population having a specified characteristic
  - e.g. income below poverty level, lacking health insurance
- Table B17001, “Poverty Status in the Past 12 Months by Sex by Age”

```
# Calculate proportion of population below federal poverty level
poverty.data <- acs.fetch(endyear = 2013,
                          geography = nh.tract,
                          variable = c("B17001_001", "B17001_002"))
poverty.pct <- divide.acs(poverty.data[,2], poverty.data[,1],
                          method = "proportion")
class(poverty.pct) # class "acs"
estimate(poverty.pct)
# percent of population below FPL for each tract
confint(poverty.pct) # 95% confidence intervals for estimates
```

Complicated measures

- Unemployment: Table B23001, “Sex by Age by Employment Status for the Population 16 Years and Over”
  - Numerator: Number unemployed
  - Denominator: Number in labor force
  - Two sexes x 13 age groups = 26 variables to combine for estimating measure for population
  - Combined numerator and denominator using “acs” version of “sum” function
    - Used “one.zero = TRUE” argument because combining large number of variables with small geographies

Unemployment measure

```
ptm <- proc.time()

# get columns from table B23001
den.var <- c("B23001_006", "B23001_013", "B23001_020", "B23001_027",
  "B23001_034", "B23001_041", "B23001_048", "B23001_055", "B23001_062",
  "B23001_069", "B23001_074", "B23001_079", "B23001_084", "B23001_092",
  "B23001_099", "B23001_106", "B23001_113", "B23001_120", "B23001_127",
  "B23001_134", "B23001_141", "B23001_148", "B23001_155", "B23001_160",
  "B23001_165", "B23001_170")
num.var <- c("B23001_008", "B23001_015", "B23001_022", "B23001_029",
  "B23001_036", "B23001_043", "B23001_050", "B23001_057", "B23001_064",
  "B23001_071", "B23001_076", "B23001_081", "B23001_086", "B23001_094",
  "B23001_101", "B23001_108", "B23001_115", "B23001_122", "B23001_129",
  "B23001_136", "B23001_143", "B23001_150", "B23001_157", "B23001_162",
  "B23001_167", "B23001_172")
unemp.data <- acs.fetch(endyear = 2013, geography = nh.tract,
   variable = append(den.var, num.var))

# aggregate 26 numerator columns and 26 denominator columns
denom <- apply(unemp.data[,den.var], MARGIN = 2, FUN = sum,
              agg.term = c("y", "denom"), one.zero = TRUE)
numer <- apply(unemp.data[,num.var], MARGIN = 2, FUN = sum,
              agg.term = c("y", "numer"), one.zero = TRUE)

# get percentage using “proportion” method
UNEMPLOYED.PCT <-  divide.acs(numer, denom, method = "proportion")

proc.time() - ptm
# about 75 seconds to run
```

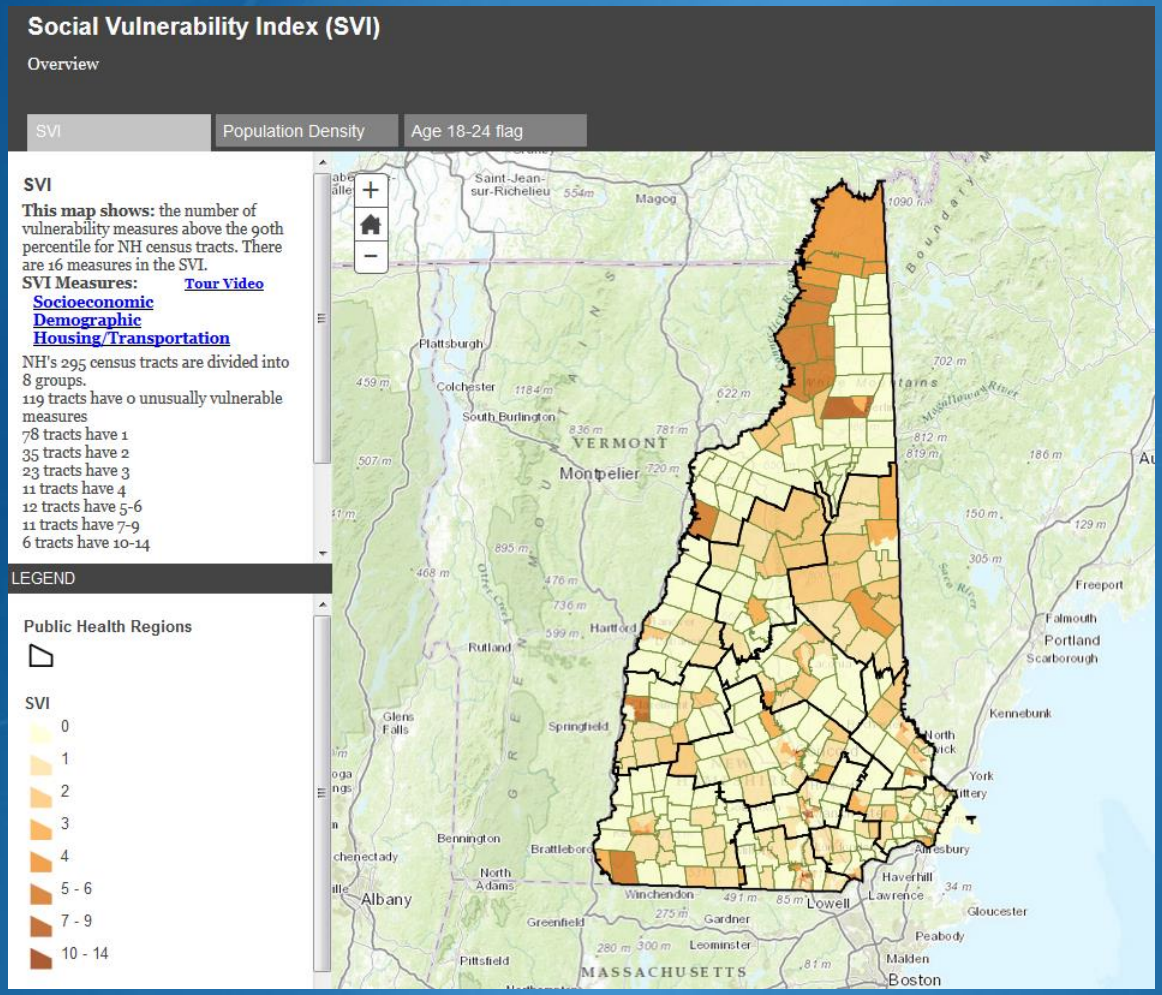
Additional processing

- Some tables were not available from Census API and had to be downloaded from American Fact Finder
  - Table B27010, “Types of Health Insurance Coverage by Age”
  - Table B18101, “Sex by Age by Disability Status”
- Used package function `read.acs` to import data
  - “sum” method would not work directly on imported object
  - Zero population tracts had to removed and geography reordered

Building the SVI index

- Index was built by combining measures using slightly modified method from Flanagan et al.
- Percentile rank computed for each tract over each variable (high rank indicates greater vulnerability)
- Each measure ranked from “most vulnerable” to “least vulnerable” across all tracts
  - E.g. per capita income ranked from lowest to highest
  - Percent of population in poverty ranked from highest to lowest
- Added results to shapefile for mapping

New Hampshire SVI online



http://nhdphs.maps.arcgis.com/home

Conclusions

- Using **`acs`** package was marked improvement over previous methods
- Package enabled automation of processes formerly done manually (e.g. uncertainty of estimates)
- Learned more about how R works (e.g. manipulation of “S4” objects)
- Learned more about ACS
- Script serves to document analysis
- Script more easily updated when new ACS data becomes available or when package changes

Postscript: PUMS data

- ACS Public Use Microdata Sample (PUMS)
  - One percent subsample of ACS data
- Provides custom crosstabs not available from ACS summary tables
- R package **`survey`** can analyze PUMS data and data from other complex surveys
- R provides complete ACS analysis toolkit

http://www.census.gov/acs/www/data_documentation/public_use_microdata_sample

http://r-survey.r-forge.r-project.org/survey/index.html

http://www.asdfree.com/

Selected resources

R Project homepage http://www.r-project.org

Introducing R
http://data.princeton.edu/R/

UCLA Institute for Digital Research and Education
http://www.ats.ucla.edu/stat/r/

RStudio development environment http://www.rstudio.com/

R for SAS, SPSS, Stata users http://r4stats.com/

Working with acs.R
http://eglenn.scripts.mit.edu/citystate/wp-content/uploads/2013/06/wpid-working_with_acs_R3.pdf

Final thought

"R has many features that SAS and SPSS lack such as: multiple data structures, a very wide range of variable selection methods, functions that optimize their output automatically for different data structures, data structure conversion tools, and workspace management functions. In short, R offers a complete and powerful programming environment rather than just a set of analytic procedures.

…

If you are a SAS or SPSS user who has happily avoided the complexities of output management, macros and matrix languages, R's added functionality may seem daunting to learn at first. On the other hand, R makes those added features so much easier to use than SAS or SPSS that you may find yourself more eager to expand your horizons. The added power of R and its free price make it well worth the effort."

-Robert A. Muenchen, *R for SAS and SPSS Users*
(Springer, 2008, p. 441)

Discussion

For more information, contact

michael.laviolette@dhhs.state.nh.us

dennis.holt@dhhs.state.nh.us