

## National Amenity Repository for the United States (NARUS) Data Collection Manual

### Introduction

This manual documents the data collection and management for the first comprehensive national database of a broad range of county-level amenities (National Amenity Repository for the United States, “NARUS”). Notice this manual primarily works on data for the year 2000 and for 2010 (although sometimes data in year 2008-2012 was used depending on the original data collection cycle).

Though some historical data were collected or even calculated, none of them was put into quality test. The manual includes information of data source, technical tips, and other notes. This manual includes three parts: (1) a detailed introduction for individual variables; (2) additional information for data source or information found but not used now, written in source order; and (3) other data-related work.

### Geographic scope and granularity

The main geographic unit of observation are counties, primarily for the contiguous states. The Contiguous states are also called lower 48 states, or conterminous states, but not continental states. State details can be found in <http://stylemanual.ngs.org/home/C/conterminous-contiguous-continental>. States Puerto Rico (PR) 72, Virgin Island (VI) 78, Guam (GU) 66, American Samoa (AS) 60, Northern Mariana Islands (MP) 69, and U.S. Minor Outlying Islands (UM) 74 are in the “territories category”, but not in the “50 states + Washington DC category”. States Alaska (AK) 02 and Hawaii (HI) 15 are in the “50 states + Washington DC category” (but not in the lower 48).

The key in bridging all data together is the FIPS code. FIPS county code is one of the ten standards. The Standard for it was FIP 6-4, and then changed into INCITS 31:2009 <http://www.census.gov/geo/www/ansi/changenotes.html>. As the administrative boundaries change, FIPS codes change too. The substantial Changes to Counties and County Equivalent Entities since 1970 can be tracked in <http://www.census.gov/geo/www/tiger/ctychng.html> (See example in [Section 1. FIPS code change](#)). A common knowledge is that in 2011, there are 3143 counties in the 50 states + DC in the U.S., and there are 3109 counties in the 48 states + DC in the U.S. To access the FIPS code data, one can go to “county and county equivalents” in <http://www.census.gov/geo/www/ansi/ansi.html> or go directly to <http://www.census.gov/geo/www/ansi/download.html>.

### Software

The file format for the data is .dta which is the native file type for Stata. However, .dta files are easily read R (and RStudio) using the read.dta command which might be expedient for formatting the information.

## Contents

|   |    |
|---|----|
| National Amenity Repository for the United States (NARUS)<br>Data Collection Manual ..... | 1  |
| Introduction.....   | 1  |
| Geographic scope and granularity.....   | 1  |
| Software .....  | 1  |
| PART 1: INDIVIDUAL VARIABLES .....  | 4  |
| 1. FIPS code change.....  | 5  |
| 2. Mean annual precipitation.....   | 5  |
| 3. Mean winter temperature .....  | 11 |
| 4. Mean summer temperature.....   | 15 |
| 5. Mean annual relative humidity.....   | 16 |
| 6. Mean July relative humidity.....   | 18 |
| 7. Mean annual heating degree days (HDD) .....  | 19 |
| 8. Mean annual cooling degree days (CDD) .....  | 21 |
| 9. Mean annual wind speed.....  | 22 |
| 10. Mean annual sunshine.....   | 23 |
| 11. Heavy fog .....   | 29 |
| 12. Percent Water Area.....   | 30 |
| 13. Coastal .....   | 31 |
| 14. Mountain peaks.....   | 31 |
| 15. Rivers.....   | 32 |
| 16. Federal land .....  | 32 |
| 17. Wilderness areas .....  | 33 |
| 18. National parks.....   | 34 |
| 19. Distance to nearest National Park .....   | 34 |
| 20. Distance to nearest State Park.....   | 35 |
| 21. Parkways (Scenic drives) and Scenic Rivers .....                                      | 36 |
| 22. Tornado.....  | 38 |
| 23. Property damage from hazard events.....   | 40 |
| 24. Seismic hazard.....   | 42 |
| 25. Number of Earthquakes .....   | 45 |
| 26. Land Cover Diversity.....   | 49 |
| 27. NPDES effluent dischargers .....  | 49 |
| 28. Landfill Waste .....  | 53 |
| 29. Superfund.....  | 55 |
| 30. Treatment, storage and disposal facilities .....                                      | 55 |
| 31. Large quantity generators of hazardous waste .....                                    | 55 |
| 32. Nuclear power plants .....  | 57 |
| 33. Coal-fired power plants .....   | 58 |

|   |     |
|---|-----|
| 34. PM2.5 .....   | 61  |
| 35. PM10 .....  | 65  |
| 36. Ozone .....   | 70  |
| 37. Sulphur dioxide .....   | 72  |
| 38. Carbon Monoxide.....  | 76  |
| 39. Nitrogen dioxide.....   | 78  |
| 40. Non-attainment areas.....                                       | 80  |
| 41. Wildfire area (Previously National Fire Plan<br>treatment)..... | 81  |
| 42. Cancer Risk .....   | 81  |
| 43. Neurological risk .....   | 86  |
| 44. Respiratory risk .....  | 86  |
| 45. Local direct general expenditures.....                          | 86  |
| 46. Local exp. for hospitals and health .....                       | 88  |
| 47. Local exp. on parks, rec. and nat. resources .....              | 88  |
| 48. Museums and historical sites .....                              | 88  |
| 49. Municipal parks (percentage of total land area) .....           | 89  |
| 50. Campgrounds and camps.....                                      | 90  |
| 51. Zoos, botanical gardens and nature parks.....                   | 90  |
| 52. Crime rate (per 100,000 persons) .....                          | 91  |
| 53. Teacher-pupil ratio .....                                       | 92  |
| 54. Local expenditure per student.....                              | 93  |
| 55. Private school to public school enrollment (%).....             | 94  |
| 56. Child mortality (per 1000 births, 1990–2000).....               | 95  |
| 57. Federal expenditure (\$ pc, non-wage, non-defense).96           |     |
| 58. Number of Airports .....  | 97  |
| 59. Number of Ports .....   | 97  |
| 60. Interstate highways (total mileage per mi <sup>2</sup> ) .....  | 98  |
| 61. Urban arterial (total milage per mi <sup>2</sup> ).....         | 99  |
| 62. Number of Amtrak stations .....                                 | 100 |
| 63. Number of urban rail stops .....                                | 100 |
| 64. Railways (total mileage per mi <sup>2</sup> ) .....             | 100 |
| 65. Number of restaurants and bars (per 1,000 people) 101           |     |
| 66. Theatres and musicals (per 1,000 people) .....                  | 101 |
| 67. Artists (per 1,000 people).....                                 | 102 |
| 68. Movie theatres (per 1,000 people).....                          | 102 |
| 69. Bowling alleys (per 1,000 people).....                          | 103 |
| 70. Amusement, recreation establishments (per 1,000<br>people) 103  |     |
| 71. Research I universities (Carnegie classification)....           | 103 |
| 72. Golf courses and country clubs .....                            | 104 |
| 73. Military areas (percentage of total land area) .....            | 105 |

|                                       |  |     |
|---------------------------------------|--|-----|
| 74.                                   | Housing stress (=1 if > 30% of hholds distressed)..    | 106 |
| 75.                                   | Persistent poverty (=1 if > 20% of pop. in poverty)    | 106 |
| 76.                                   | Retirement destination (=1 if growth retirees > 15%)   | 107 |
| 77.                                   | Distance (km) to the nearest urban center.....         | 107 |
| 78.                                   | Incr. distance to a metropolitan area of any size..... | 107 |
| 79.                                   | Incr. distance to a metro area > 250,000.....          | 107 |
| 80.                                   | Incr. distance to a metro area > 500,000.....          | 108 |
| 81.                                   | Incr. distance to a metro area > 1.5 million .....     | 108 |
| 82.                                   | Gambling .....   | 108 |
| 83.                                   | Land Grant University .....                            | 108 |
| PART 2: DATA SOURCE .....             |  | 109 |
| 1.                                    | NOAA-NCDC.....   | 109 |
| 2.                                    | ICPSR .....  | 109 |
| 3.                                    | NOAA-SEAD .....  | 109 |
| 4.                                    | ESRI.....  | 109 |
| 5.                                    | USDI-NPS.....  | 109 |
| 6.                                    | USGS-NA .....  | 109 |
| 7.                                    | EPA-TRI .....  | 109 |
| 8.                                    | USDOE-INSC.....  | 111 |
| 9.                                    | EPA-AQS.....   | 111 |
| 10.                                   | EPA-NATA .....   | 111 |
| 11.                                   | COG.....   | 111 |
| 12.                                   | CBP.....   | 113 |
| 13.                                   | Census.....  | 114 |
| 14.                                   | USDS-ERS .....   | 114 |
| 15.                                   | CDC-NCHS .....   | 114 |
| 16.                                   | CCIHE .....  | 114 |
| 17.                                   | PRAO-JIE09 .....                                       | 114 |
| PART 3: OTHER DATA-RELATED WORK ..... |  | 114 |
| 1.                                    | Compare LIST with D3’s list .....                      | 114 |
| 2.                                    | Overview of Existing Sustainability Indicators.....    | 114 |
| APPENDIX.....                         |  | 114 |
| 1: Status track table.....            |  | 114 |

## PART 1: INDIVIDUAL VARIABLES

### Population-weighted centroid

You can download the file calculated by the census bureau itself from the following website:

<http://www.census.gov/geo/reference/centersofpop.html>

### Population at the block group level

It is important to have the population at the block group level, since these figures might be used to calculate a given variable population-weighted mean at the county. To obtain these, one needs to download total population at the block group level. This can be downloaded from the following website:

[http://www2.census.gov/census\\_2010/04-Summary\\_File\\_1/](http://www2.census.gov/census_2010/04-Summary_File_1/)

In order to process these files, one needs to import these files into Microsoft Access. There is a step-by-step guide on how to accomplish this:

[http://www2.census.gov/census\\_2010/04-Summary\\_File\\_1/0HowToUseMSAccessWithSummaryFile1.pdf](http://www2.census.gov/census_2010/04-Summary_File_1/0HowToUseMSAccessWithSummaryFile1.pdf)

It is important to download the files per state and not the nation-wide files. The nation-wide files have no information regarding the states, so we have to download the files per state and create population tables per each state of interest.

Follow the previous guide in order to export Excel files of population by block group per each state. Once this is done you will be able to import these Excel files into Stata and then append them into a single dta file.

### Place-County relationship file

In some instances it will be useful to have a table that contains the relationship information between places (i.e. cities, towns, townships, etc.) and counties. In order to do so, I contacted the Census Bureau and they provided the following website:

<http://mcdr.missouri.edu/websas/geocorr12.html>

In this website one can select all the states, and D.C., from the Contiguous U.S. and then select 'Place' as "Source" and 'County' as "Target". Then I selected 'Population (2010 census)' as the weighting variable. After doing this you can 'Run the request' and you will obtain a csv file ready to download after some minutes.

Now we can import the file into Stata. The first and second rows should be the variable names and variable labels, respectively.

```
> rename v1 statecode
> rename v2 placefp
> rename v3 countycode
> rename v4 state
> rename v5 county
> rename v6 placename
> rename v7 pop10
> rename v8 afact
> label variable statecode "State code"
> label variable placefp "placefp"
> label variable countycode "County code"
> label variable state "State"
> label variable county "County"
> label variable placename "Place Name"
```

```
> label variable pop10 "Total Pop, 2010 census"
> label variable afact "place to county allocation factor"
> drop if statecode=="state"
> gen lengthcounty = length(county)
> gen county2 = substr(county,1,lengthcounty-3)
> gen lengthplace = length(placename)
> gen placename2 = substr(placename,1,lengthplace-4)
> drop county placename lengthcounty lengthplace
> rename county2 county
> rename placename2 placename
> label variable county "County"
> label variable placename "Place Name"
> order placefp , last
> order countycode , first
> order statecode , first
> order placename , first
> order county , first
> order state , first
```

## 1. FIPS code change

The major change between 2000 and 2010 is that Clifton Forge (independent) city, Virginia (51-560) was changed to town status and added to Alleghany County (51-005) effective July 1, 2001. A solution is to add 51-560 information to 51-005. To realize it, one can create a new variable FIPS2, and change 51560 into 51005 in FIPS2. Therefore, there are two 51005 in FIPS2 with different population. Then use “collapse by FIPS2 [fw = pop]” which means using pop as frequent weight to merge.

```
> collapse by FIPS2 [fw = pop]
```

| FIPS  | FIPS2 | pop  |
|-------|-------|------|
| 51560 | 51005 | pop1 |
| 51005 | 51005 | pop2 |

## 2. Mean annual precipitation

### 1981-2010 climate normals

NOAA’s National Climate Data Center (NCDC) provides the 1981-2010 Climate Normals in the following website:

<http://www.ncdc.noaa.gov/oa/climate/normal/usnormals.html>

This website also offers FAQs, which could be useful if one has little experience working with climatological variables. The readme file can be found in the website: <http://www1.ncdc.noaa.gov/pub/data/normal/1981-2010/readme.txt>.

It is vital to read the readme.txt file. For instance, in this file, it is mentioned that the units in which precipitation is stored are hundredth of inches and whole degrees Fahrenheit for heating and cooling degree days (expect high precision files, these are stored in hundredths of degrees Fahrenheit).

|                                       |                      |
|---------------------------------------|----------------------|
| Mean precipitation                    | hundredths of inches |
| Mean annual heating degree days (HDD) | °F                   |
| Mean annual cooling degree days (CDD) | °F                   |

The information can be downloaded from the following website:

<http://www1.ncdc.noaa.gov/pub/data/normals/1981-2010/>

After downloading the information, it is necessary to import it into Stata.

Run: File\Import\Text data created by a spreadsheet; a window will appear.

Load the .txt file that you downloaded and keep all the default settings. It is very important not to open the txt file and edit it, doing so will add some strange characters before the first record and you may lose this first observation.

```
> insheet using "E:\Research\NOAA-NCDC\data\1981-2010Normals\ann-prcp-normal.txt", clear
```

We need to split this single imported variable into “stationid”, “mean precipitation” and “flag.” In order to do this, we need to run the command:

```
> gen length = length(v1)
> gen stationid = substr(v1,1,11)
> gen v2 = substr(v1,12,length)
> drop length v1
> gen length = length(v2)
> sum length
> gen precipitation = substr(v2,1,12)
> gen flag = substr(v2,13,.)
> destring precipitation , replace
> drop v2 length
> replace precipitation = precipitation/100
> label variable stationid "Station ID"
> label variable precipitation "Annual precipitation in inches, 1981-2010"
```

There may be negative values in the data. Read the readme.txt file in order to understand the syntax. According to the normal 1981-2010 normals readme.txt file, the negative values have the following meaning:

|       |  |
|-------|--|
| -9999 | missing or insufficient data; values cannot be computed  |
| -8888 | date not defined (e.g. February 30, September 31) - used in daily files to achieve fixed-length records                  |
| -7777 | a non-zero value that would round to zero, for variables bound by zero.  |
| -6666 | parameter undefined; used in precipitation/snowfall/snow depth percentiles when number of nonzero values is insufficient |
| -5555 | parameter not available because it was inconsistent with another parameter   |

Run the following command to identify how many observations have negative precipitation values:

```
> sum precipitation
```

If the minimum value is negative, then we need to run commands to replace the values accordingly. For instance, since “-7777” means that the values is close enough to zero, we can round this value to zero.

```
> replace precipitation = 0 if precipitation == -7777
```

Then we can save this file. Remember that this file contains station ID, normal precipitation and flag code. It does not contain location information.

```
> save "E:\Research\NOAA-NCDC\Precipitation\precipitation_1981-2010.dta"
```

Now, there is another methodology for importing data. One could use a dictionary file. In order to create a dictionary file, one needs to know the format of the .txt file, i.e. the column in which each variable starts, the length of the variable, and the type of the variable. The name is not crucial, since one can assign the name that one thinks is best suited.

The syntax of each row in the dictionary file is the following:

`_column("a") str"b" "varname" %"#c "varlabel"`

Where:

- "a" column in which the variable starts
- "b" length of the string variable(omit if variable is a number)
- # length of the variable. If the variable is string just type the total length of the variable, if the variable is numeric, type the total length followed by a dot and the number of decimals (e.g. 8.4)
- c s if variable is string, f if variable is numeric

For instance, the dictionary file to import the location information of all the stations used for preparing the climate normals is the following:

```
infile dictionary {
_column(1) str11 stationid %11s "Station ID"
_column(13) latitude %8.4f "Latitude"
_column(22) longitude %9.4f "Longitude"
_column(32) elev %6.1f "Elevation"
_column(39) str2 state %2s "State"
_column(42) str30 name %30s "Station name"
_column(73) str3 gsnflag %3s "GSN Flag"
_column(77) str3 hcnflag %3s "HCN Flag"
_column(81) str5 WMOID %5s "WMO ID"
}
```

In order to create a dictionary file, open the do-file editor (Window\Do-file Editor\New Do-File Editor), type the syntax of the dictionary and save it as a .dct file. Once the dictionary file is created, run 'File\Import\Text data in fixed format with a dictionary'. The location information can be downloaded from:

<http://www1.ncdc.noaa.gov/pub/data/normals/1981-2010/station-inventories/>. The file is named 'allstations.txt'.

Then load the .txt file and the dictionary file and the .txt file should be imported and formatted into a .dta file. Also, instead of importing the txt file by using the user-friendly windows, we can run the following command, in which we will have to specify the txt path and dct path.

```
> infile using "E:\Research\NOAA-NCDC\station_location\normals\allstations.dct", using("E:\Research\NOAA-NCDC\station_location\normals\allstations.txt") clear
> save "E:\Research\NOAA-NCDC\station_location\normals\allstations.dta", replace
```

Then, since we had no long/lat information of the stations in the precipitation\_1981-2010.dta file, we need to merge this file with the allstations.dta file. To perform this, we need to open the precipitation file and then run: Data\Combine datasets\Merge two datasets. It is important that both files have the same variable name for the station id; specify this variable name as key variable and load the stata file that contains the information of latitude and longitude of the stations. Once the precipitation file is merged with this one, you can drop the dispensable variables (e.g. hcnflag). Alternatively, you can run the following command:

```
> use "E:\Research\NOAA-NCDC\Precipitation\precipitation_1981-2010.dta", clear
> merge 1:1 stationid using "E:\Research\NOAA-NCDC\station_location\normals\allstations.dta"
> drop elev name state gsnflag hcnflag WMOID _merge
> drop if missing(precipitation)
> label variable latitude "Latitude in decimal degrees"
> label variable longitude "Longitude in decimal degrees"
```

In order to drop all the stations that are outside the Conterminous U.S., you can run the following commands:

```
> drop if(latitude<23)
> drop if(latitude>50)
> drop if(longitude<-126)
> drop if(longitude>-65)
```

There may be some duplicate information from stations who share the same location but have slightly different records. Thus, one needs to collapse the information on longitude and latitude by running the following commands:

```
> collapse (mean) precipitation , by(longitude latitude)
> label variable precipitation "Annual precipitation in inches, 1981-2010"
```


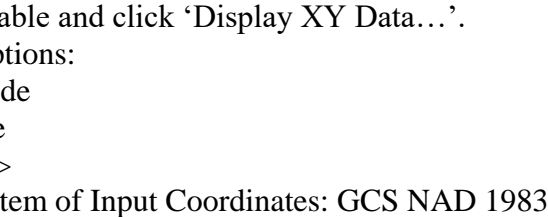
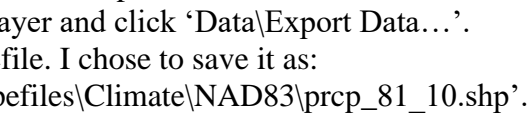
Then we can save this file.

```
> save "E:\Research\NOAA-NCDC\Precipitation\precipitation_1981-2010_geo.dta"
```

Then we can export this file as a xls file in order to import it into ArcGIS. Since there is no information regarding the geographic coordinate system, I am assuming that the GCS used is NAD 1983.

```
> export excel using "E:\Research\NOAA-NCDC\Precipitation\precipitation_1981-2010_geo.xls", firstrow(variables)
```

Once we have exported the dta file as an excel file, we need to do the following to import into ArcGIS.

- (1) Open ArcMap.
- (2) Click on 'Add Data' 
- (3) Add the first sheet of the excel file. In this case, this file is called 'precipitation\_1981-2010\_geo.xls'.
- (4) Right click the added table and click 'Display XY Data...'.  

- (5) Select the following options:
  - a. X field: longitude
  - b. Y field: latitude
  - c. Z field: <None>
  - d. Coordinate System of Input Coordinates: GCS NAD 1983
- (6) Right click the added layer and click 'Data\Export Data...'.  

- (7) Then save it as a shapefile. I chose to save it as:  
'E:\Research\GIS\Shapefiles\Climate\NAD83\prcp\_81\_10.shp'.

Now that we have plotted the precipitation information spatially, we need to calculate the precipitation at the county population-weighted centroid. I am using the ordinary kriging in order to do so, since according to Anselin & Le Gallo<sup>1</sup>, kriging provides the best fit and most reasonable parameter signs and magnitudes when compared against Thiessen polygons, inverse distance weighting and spline interpolation. Since they used ordinary kriging, I am also using ordinary kriging.

Also, since the shapefile is currently not projected, we need to project it prior to running the kriging. I recommend to project it into an equidistant projection, in order to account more appropriately for the distances between climate stations and county population-weighted centroids. I'm using the following projection in R:

```
"+proj=eqdc +lat_0=39 +lon_0=-96 +lat_1=33 +lat_2=45 +x_0=0 +y_0=0 +datum=NAD83 +units=m +no_defs +ellps=GRS80 +towgs84=0,0,0"
```

Universal kriging is used, using long lat as explanatory variables in the linear model. After running the universal kriging in R, it can be exported as a txt file and then import it into Stata.

It is very important to choose explanatory variables that are significant in determining the dependent variable trend. For instance, in the case of precipitation, I decided to run the kriging first on a binomial expression of the location, i.e. ' $x+y+I(x^2)+I(y^2)+I(x*y)$ '. The expression for modeling the variogram looks this way:

---

<sup>1</sup> Luc Anselin & Julie Le Gallo (2006): Interpolation of Air Quality Measures in Hedonic House Price Models: Spatial Aspects, *Spatial Economic Analysis*, 1:1, 31-52.



```
> vt.preci.uk <- variogram(precipitation ~ x+y+I(x^2)+I(x*y), preci.clim, cloud=F)
> plot(vt.preci.uk)
> vt.fit.preci.uk <- fit.variogram(vt.preci.uk, vgm(psill=80,model="Gau",range=850000,nugget=30))
> plot (vt.preci.uk, vt.fit.preci.uk)
> vt.fit.preci.uk
```

Then the expression for assessing the significance of the  $\beta$  coefficients is the following:

```
> preci.g.uk <- gstat(formula=precipitation ~ x+y+I(x^2)+I(x*y), data=preci.clim, model = vt.fit.preci.uk)
> predict(preci.g.uk, newdata=preci.clim[1,], BLUE=TRUE, debug=32)
```

The output of the last line of code is very large, however the one we are interested in is the one that starts with '# beta'. It looks like this:

```
# beta:
Vector: dim: 6
21.6069472 5.78348059e-006 2.62892028e-006 9.6737998e-012 -6.41319907e-012 -1.49088157e-011
# Cov(beta), (X'C-1 X)-1:
Matrix: 6 by 6
row 0: 26.2924369 2.99343678e-007 1.24258467e-006 -4.06692992e-012 -7.55153886e-012 -8.89926995e-013
row 1: 2.99343678e-007 3.82485765e-012 5.78219573e-013 1.54458561e-019 -6.89540059e-019 -1.11369249e-018
row 2: 1.24258467e-006 5.78219573e-013 7.47899885e-012 -6.86496836e-019 -1.90160296e-019 -4.62847089e-019
row 3: -4.06692992e-012 1.54458561e-019 -6.86496836e-019 1.58934974e-024 3.13953579e-025 9.00968595e-026
row 4: -7.55153886e-012 -6.89540059e-019 -1.90160296e-019 3.13953579e-025 6.54657025e-024 1.23769273e-024
row 5: -8.89926995e-013 -1.11369249e-018 -4.62847089e-019 9.00968595e-026 1.23769273e-024 3.5834682e-024
```

Then, we can calculate the t-values for the coefficients; these results will be taken into account to decide if some explanatory variables should not be taken into account or if the model looks adequate enough.

```
preci.x.coef <- 5.78348059e-006
preci.y.coef <- 2.62892028e-006
preci.x2.coef <- 9.6737998e-012
preci.y2.coef <- -6.41319907e-012
preci.xy.coef <- -1.49088157e-011

preci.x.se <- 3.82485765e-012^0.5
preci.y.se <- 7.47899885e-012^0.5
preci.x2.se <- 1.58934974e-024^0.5
preci.y2.se <- 6.54657025e-024^0.5
preci.xy.se <- 3.5834682e-024^0.5

preci.x.coef/preci.x.se
preci.y.coef/preci.y.se
preci.x2.coef/preci.x2.se
preci.y2.coef/preci.y2.se
preci.xy.coef/preci.xy.se
```

In this case, 'y' seemed not be significant, thus I decided to drop 'y<sup>2</sup>', to assess if by dropping this variable, 'y' might become significant. I decided not to drop 'y' because that would force the model to have the y influence set to zero. After running the model as 'x+y+I(x<sup>2</sup>)+I(x\*y)', the significance of y did not improve, thus I decided to use the first model. Using this model, I performed the universal kriging to obtain the mean precipitation on the county population-weighted centroids.

### 1971-2000 climate normals

In the case of 1971-2000 Climate Normals, this data can be downloaded from the following website:

[http://cdo.ncdc.noaa.gov/cgi-bin/climatenormals/climatenormals.pl?directive=prod\\_select2&prodtype=CLIM81&subnum=](http://cdo.ncdc.noaa.gov/cgi-bin/climatenormals/climatenormals.pl?directive=prod_select2&prodtype=CLIM81&subnum=)

Then we need to select the state and download the data by state.

Once we have downloaded the .txt files, we need to create dictionary files in order to easily import the data into Stata. The dictionary files have the following syntax:

#### station\_location.dct

```
infile dictionary{
_column(1) str3 station_number %3s "Station number"
_column(8) str6 COOPID %6s "COOP Station ID"
_column(32) str28 station_name %28s "Station name"
_column(67) str2 lat_deg %2s "Latitude degrees"
_column(70) str2 lat_min %2s "Latitude minutes"
_column(75) str3 long_deg %3s "Longitude degrees"
_column(79) str2 long_min %2s "Longitude minutes"
}
```

#### precipitation.dct

```
infile dictionary{
_column(1) str3 station_number %3s "Station number"
_column(5) str28 station_name %28s "Station name"
_column(106) str7 precipitation %7s "Mean annual precipitation in inches, 1971-2000"
}
```

Run the following commands to generate the station location dta file:

```
> infile using "E:\Research\NOAA-NCDC\data\1971-2000Normals\station_location.dct", using("E:\Research\NOAA-NCDC\data\1971-2000Normals\ALnorm.txt") clear
> drop if missing(lat_min)
> drop if lat_deg=="_"
> drop if COOPID=="COOPID"
> destring , replace
> gen state = "AL"
> gen latitude = lat_deg+lat_min/60
> gen longitude = long_deg+long_min/60
> drop lat_deg lat_min long_deg long_min
> label variable latitude "Latitude in decimal degrees"
> label variable longitude "Longitude in decimal degrees"
> save "E:\Research\NOAA-NCDC\data\1971-2000Normals\AL_stations.dta"
```

Then we need to generate the precipitation dta files.

```
> infile using "E:\Research\NOAA-NCDC\data\1971-2000Normals\precipitation.dct", using("E:\Research\NOAA-NCDC\data\1971-2000Normals\ALnorm.txt") clear
> drop if missing(precipitation)
> gen length = length(station_name)
> drop if substr(station_name,length-2,length)=="MAX"
> drop if station_name=="MEAN"
> drop if station_name=="MTN"
> drop if station_number=="NO."
> drop if station_number=="_"
> drop if substr(station_name,length-2,length)=="CDD"
> drop if substr(station_name,length-2,length)=="HDD"
> drop if substr(station_name,length-3,length)=="CDD*"
> drop if substr(station_name,length-3,length)=="HDD*"
> drop if substr(station_name,length-3,length)=="MEAN"
> drop if substr(station_name,length-3,length)=="YEAR"
> drop if substr(station_name,length-5,length)=="MEDIAN"
> destring , replace
> drop length
> merge 1:1 station_number using "E:\Research\NOAA-NCDC\data\1971-2000Normals\AL_stations.dta"
> save "E:\Research\NOAA-NCDC\data\1971-2000Normals\AL_precipitation.dta"
```

Then we need to append the precipitation dta files together, and finally import them into ArcGIS.

```
> use "E:\Research\NOAA-NCDC\data\1971-2000Normals\AL_precipitation.dta", clear
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\AR_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\AZ_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\CA_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\CO_precipitation.dta"
```

```
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\CT_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\DE_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\FL_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\GA_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\IA_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\ID_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\IL_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\IN_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\KS_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\KY_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\LA_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MA_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MD_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\ME_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MI_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MN_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MO_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MS_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MT_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NC_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\ND_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NE_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NH_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NJ_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NM_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NV_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NY_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\OH_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\OK_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\OR_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\PA_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\RI_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\SC_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\SD_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\TN_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\TX_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\UT_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\VA_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\VT_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WA_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WI_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WV_precipitation.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WY_precipitation.dta"
> drop station_number_merge
> order COOPID state station_name precipitation , first
> save "E:\Research\NOAA-NCDC\Precipitation\precipitation_1971-2000_geo.dta"
> export excel using "E:\Research\NOAA-NCDC\Precipitation\precipitation_1971-2000_geo.xls", firstrow(variables)
```

Finally, see final section of 1981-2010 climate normals Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean winter temperature at the county population-weighted centroids.

### 3. Mean winter temperature

According to Sinha & Cropper<sup>2</sup>, using mean winter temperature and summer winter temperature (Section 4) is advantageous since they capture seasonality, which annual heating and cooling degree days and temperature bins do not. However, we are also considering mean annual heating degree days (Section 6) and mean annual cooling degree days (Section 7) since these are benchmark variables.

Note: There are two datasets with daily information: the Daily Summaries from the Daily Global Historical Climatology Network (GHCN-Daily) and the Global Summary of the Day dataset. Both of them can be found

---

<sup>2</sup> Paramita Sinha & Maureen L. Cropper (2013): The Value of Climate Amenities: Evidence from US Migration Decisions. National Bureau of Economic Research.

in the following website: <http://www.ncdc.noaa.gov/cdo-web/datasets>. However, the first one, i.e. Daily Summaries, only contains precipitation, snowfall, snow depth, and maximum and minimum daily temperature, whereas the second one, i.e. Global Summary of the Day, contains mean temperature, mean dew point, mean visibility, mean wind speed, among others. Thus, the Global Summary of the Day dataset can be used for obtaining the mean winter temperature (Section 3), mean summer temperature (Section 4), mean annual relative humidity (Section 5), mean wind speed (Section 8) and heavy fog (Visibility – Section 10).

Daily meteorological data was downloaded from the Global Surface Summary of the Day Dataset, from NOAA. <ftp://ftp.ncdc.noaa.gov/pub/data/gsod/>

The data has the following units (reference: <ftp://ftp.ncdc.noaa.gov/pub/data/gsod/readme.txt>):

Mean temperature (degrees Fahrenheit)

Mean dew point (degrees Fahrenheit)

Mean sea level pressure (mb)

Mean station pressure (mb)

Mean visibility (miles)

Mean wind speed (knots)

Maximum sustained wind speed (knots)

Maximum wind gust (knots)

Maximum temperature (degrees Fahrenheit)

Minimum temperature (degrees Fahrenheit)

Precipitation amount (inches)

Snow depth (inches)

Indicator for occurrence of:

- Fog
- Rain/Drizzle
- Snow/Ice Pellets
- Hail
- Thunder
- Tornado/Funnel Cloud

In this website, one can download data per year; however, the data is downloaded in .gz format. In order to decompress this file, you need to download a program such as 7zip. After decompressing a year file, you should get more than 8000 files (one per station) in .gz format. This is a compressed file too, so you need to select them all and decompress them again. Then, you'll get .op files, which are some sort of tab delimited files. Now, since we have a single file per station, it is necessary to merge all these files into one single .txt file. In order to do this, you need to open the “Command Prompt” program, and then change the directory until you get to the folder in which you have the .op files you want to merge. For instance, the initial directory may be “C:\Users\diegohornamunoz” and you stored the files in the directory “E:\Research\NOAA-NCDC\data\Global\_summary\_of\_the\_day”.

In order to move to this subfolder in a different drive you need to type:

```
> E:
> cd Research
> cd NOAA-NCDC
> cd data
> cd GSOD
> cd 1981
```

To move up to a subfolder you need to type:

```
> cd..
```

Then, once you are in the directory where the .op files are stored, you need to type:

```
> copy *.op 1981.txt
```

By doing this, you will have created a new .txt file which contains all the information from the op files in that folder; however, this txt file also contains the header of each variable name as a row and therefore, when imported into Stata the imported file will have some observations with the name of the variables.

To import the file into Stata, a dictionary file must be created. Since there will be some observations that will have string variables, it is important to state that all the variables are strings in the dictionary file. The syntax for the dictionary file is the following:

```
infile dictionary{
_column(1) str6 STN %6s "STN"
_column(8) str5 WBAN %5s "WBAN"
_column(15) str8 YEARMODA %8s "Year(4) Month(2) Day(2)"
_column(25) str6 temp %6s "Temperature"
_column(33) str1 tempcode %1s "Temperature code"
_column(36) str6 DewP %6s "Dew point"
_column(44) str1 DewPcode %1s "Dew point code"
_column(47) str6 SLP %6s "Sea Level Pressure"
_column(55) str1 SLPcode %1s "Sea Level Pressure code"
_column(58) str6 STP %6s "Station Pressure"
_column(66) str1 STPcode %1s "Station Pressure code"
_column(68) str6 visib %6s "Visibility"
_column(76) str1 vis_code %1s "Visibility code"
_column(78) str6 WDSP %6s "Mean wind speed"
_column(86) str1 WDSPcode %1s "Mean wind speed code"
_column(88) str6 MXSPD %6s "Max wind speed"
_column(96) str5 Gust %5s "Gust"
_column(104) str5 MAX %5s "Max temperature"
_column(112) str5 MIN %5s "Min temperature"
_column(118) str6 Prcp %6s "Precipitation amount"
_column(126) str5 Snow %5s "Snow Depth"
_column(133) str6 FRSHTT %6s "Code for occurrence of"
}
```

We can import the file by running the following command:

```
> infile using "E:\Research\NOAA-NCDC\data\GSOD\GSOD.dct", using("E:\Research\NOAA-NCDC\data\GSOD\1981.txt")
clear
```

Once a .txt file is imported, we can drop the observations that are actually headers from each individual dataset by running the following command:

```
> drop if (YEARMODA=="YEARMODA")
```

Since all the data was imported as string, now we have to convert it into numerical variables. To do this we can run the following command:

```
> destring , replace
```

Then we can save this file.

```
> save "E:\Research\NOAA-NCDC\data\GSOD\1981.dta"
```

And then repeat for all annual files. Once we have all the annual files from 1981 to 2010, we can append them all together.

```
> use "E:\Research\NOAA-NCDC\data\GSOD\1981.dta", clear
> append using "E:\Research\NOAA-NCDC\data\GSOD\1982.dta" "E:\Research\NOAA-NCDC\data\GSOD\1983.dta"
"E:\Research\NOAA-NCDC\data\GSOD\1984.dta" "E:\Research\NOAA-NCDC\data\GSOD\1985.dta" "E:\Research\NOAA-
NCDC\data\GSOD\1986.dta" "E:\Research\NOAA-NCDC\data\GSOD\1987.dta" "E:\Research\NOAA-NCDC\data\GSOD\1988.dta"
"E:\Research\NOAA-NCDC\data\GSOD\1989.dta" "E:\Research\NOAA-NCDC\data\GSOD\1990.dta" "E:\Research\NOAA-
NCDC\data\GSOD\1991.dta" "E:\Research\NOAA-NCDC\data\GSOD\1992.dta" "E:\Research\NOAA-NCDC\data\GSOD\1993.dta"
"E:\Research\NOAA-NCDC\data\GSOD\1994.dta" "E:\Research\NOAA-NCDC\data\GSOD\1995.dta" "E:\Research\NOAA-
NCDC\data\GSOD\1996.dta" "E:\Research\NOAA-NCDC\data\GSOD\1997.dta" "E:\Research\NOAA-NCDC\data\GSOD\1998.dta"
"E:\Research\NOAA-NCDC\data\GSOD\1999.dta" "E:\Research\NOAA-NCDC\data\GSOD\2000.dta" "E:\Research\NOAA-
NCDC\data\GSOD\2001.dta" "E:\Research\NOAA-NCDC\data\GSOD\2002.dta" "E:\Research\NOAA-NCDC\data\GSOD\2003.dta"
"E:\Research\NOAA-NCDC\data\GSOD\2004.dta" "E:\Research\NOAA-NCDC\data\GSOD\2005.dta" "E:\Research\NOAA-
NCDC\data\GSOD\2006.dta" "E:\Research\NOAA-NCDC\data\GSOD\2007.dta" "E:\Research\NOAA-NCDC\data\GSOD\2008.dta"
"E:\Research\NOAA-NCDC\data\GSOD\2009.dta" "E:\Research\NOAA-NCDC\data\GSOD\2010.dta"
> keep STN WBAN YEARMODA temp DewP visib WDSF
> save "E:\Research\NOAA-NCDC\data\GSOD\1981-2010.dta"
```

It is important to save this dta file, since this one will be used lately to calculate the ‘Mean summer temperature’ (Section 4), ‘Mean relative humidity’ (Section 5), ‘Mean wind speed’ (Section 8), and ‘Heavy fog’ (Section 10). Now that we have saved this file, we can begin processing this dta file.

Using this dataset, one can drop the days that are not within the winter season (i.e. December 21 and March 19). To do this, one can run the following commands:

```
> use "E:\Research\NOAA-NCDC\data\GSOD\1981-2010.dta" , clear
> keep STN WBAN YEARMODA temp
> gen year = floor(YEARMODA/10000)
> gen month = floor((YEARMODA-year*10000)/100)
> gen day = (YEARMODA-year*10000)-month*100
> drop YEARMODA
> keep if (month==12|month==1|month==2|month==3)
> drop if (month==12&day<21)
> drop if (month==3&day>19)
> drop if missing(temp)
```

Then, we can collapse the information in order to obtain the mean temperature during this period. I also dropped the records that had less than 66 days of information per period, since these records would have less than 75% of the supposedly recorded period (89 days).

```
> collapse (mean) temp (count) day , by (STN WBAN year)
> drop if day < 66
> collapse temp , by (STN WBAN)
> label variable temp "Mean winter temperature in degrees Fahrenheit, 1981-2010"
> rename temp wintertemp
> sort STN WBAN
> save "E:\Research\NOAA-NCDC\Temperature\wintertemp_1981-2010.dta"
```

This data has no information regarding location of the meteorological stations. However, this can be downloaded from a different folder within the NOAA-NCDC ftp site.

<ftp://ftp.ncdc.noaa.gov/pub/data/inventories/ISH-HISTORY.TXT>

Once this data is downloaded, a dictionary file was created in order to import this data. The syntax for the dictionary file is the following:

```
infile dictionary{
  _column(1) str6 STN %6s "Air Force Datsav3 station number"
  _column(8) str5 WBAN %5s "WBAN"
  _column(14) str30 Station_Name %30s "Station name"
  _column(44) str3 Country_ID1 %3s "WMO historical country ID"
  _column(47) str3 Country_ID2 %3s "FIPS country ID"
  _column(50) str3 State %3s "State for US stations"
  _column(53) str6 CALL %6s "ICAO call sign"
  _column(59) str7 latitude %7s "Latitude in thousandths of decimal degrees"
```

```
_column(66) str8 longitude %8s "Longitude in thousandths of decimal degrees"  
_column(74) str10 Elevation %10s "Elevation in tenths of meters"  
_column(84) str9 Begin %9s "Beginning Period of Record"  
_column(93) str8 End %8s "Ending Period of Record"  
}
```

We need to import this txt file and then merge it together with the wintertemp\_1981-2010.dta file. Also, it's better to have the latitude and longitude in degrees, rather than thousandths of degrees, so replace these variables.

```
> infile using "E:\Research\NOAA-NCDC\station_location\GSOD\ish-history-stations.dct", using("E:\Research\NOAA-NCDC\station_location\GSOD\ish-history-stations.txt") clear  
> destring , replace  
> keep STN WBAN latitude longitude  
> drop if latitude==--99999|longitude==--99999  
> replace latitude = latitude/1000  
> replace longitude = longitude/1000  
> drop if missing(latitude)|drop if missing(longitude)  
> label variable latitude "latitude in decimal degrees"  
> label variable longitude "longitude in decimal degrees"  
> save "E:\Research\NOAA-NCDC\station_location\GSOD\ish-history-stations.dta", replace  
> use "E:\Research\NOAA-NCDC\Temperature\wintertemp_1981-2010.dta", clear  
> merge 1:1 STN WBAN using "E:\Research\NOAA-NCDC\station_location\GSOD\ish-history-stations.dta"  
> keep if _merge==3  
> drop _merge
```

Now, once the datafiles are merged, it is necessary to make sure that this data is in order, so I run the following command:

```
> sort STN WBAN
```

After performing this one can drop the observations that are not within the Coterminous U.S. Run the following commands:

```
> drop if(latitude<23)  
> drop if(latitude>50)  
> drop if(longitude<-126)  
> drop if(longitude>-65)
```

Then we can save this file.

```
> save "E:\Research\NOAA-NCDC\Temperature\wintertemp_1981-2010_geo.dta"
```

Then we can export this file as a xls file in order to import it into ArcGIS. Since there is no information regarding the geographic coordinate system, I am assuming that the GCS used is NAD 1983.

```
> export excel using "E:\Research\NOAA-NCDC\Temperature\wintertemp_1981-2010_geo.xls", firstrow(variables)
```

Finally, see final section of Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean winter temperature at the county population-weighted centroids.

## 4. Mean summer temperature

Using information from the GSOD data (See Mean Winter Temperature, Section 3), one can drop the days that are not within the summer season (i.e. June 21 and September 21). To do this, one can run the following commands:

```
> use "E:\Research\NOAA-NCDC\data\GSOD\1981-2010.dta", clear
> keep STN WBAN YEARMODA temp
> gen year = floor(YEARMODA/10000)
> gen month = floor((YEARMODA-year*10000)/100)
> gen day = (YEARMODA-year*10000)-month*100
> drop YEARMODA
> keep if (month==6|month==7|month==8|month==9)
> drop if (month==6&day<21)
> drop if (month==9&day>21)
> drop if missing(temp)
```

Then, we can collapse the information in order to obtain the mean temperature during this period.

```
> collapse (mean) temp (count) day , by (STN WBAN year)
> drop if day < 66
> collapse temp , by (STN WBAN)
> label variable temp "Mean summer temperature in degrees Fahrenheit, 1981-2010"
> rename temp summertemp
> sort STN WBAN
> save "E:\Research\NOAA-NCDC\Temperature\summertemp_1981-2010.dta"
> merge 1:1 STN WBAN using "E:\Research\NOAA-NCDC\station_location\GSOD\ish-history-stations.dta"
> keep if _merge==3
> drop _merge
> sort STN WBAN
> drop if (latitude<23)
> drop if (latitude>50)
> drop if (longitude<-126)
> drop if (longitude>-65)
> save "E:\Research\NOAA-NCDC\Temperature\summertemp_1981-2010_geo.dta"
```

After running this we can merge the `summertemp_1981-2010_geo.dta` file and the `wintertemp_1981-2010_geo.dta` file in order to compare the summer and winter temperatures and drop if any observation has a mean summer temperature lower than the mean winter temperature. Obviously, we expect the summer temperature to be higher than the winter temperature for all the stations, if there is any station with a winter temperature higher than the summer temperature, this stations needs to be dropped in both datasets.

```
> use "E:\Research\NOAA-NCDC\Temperature\summertemp_1981-2010_geo.dta", clear
> merge 1:1 STN WBAN using "E:\Research\NOAA-NCDC\Temperature\wintertemp_1981-2010_geo.dta"
> drop if missing(summertemp)|missing(wintertemp)
> keep if summertemp<wintertemp
```

If there are no records, then this means that there are no stations with mean summer temperature lower than the mean winter temperature. In this case there are no records, thus no observation needs to be dropped.

```
> use "E:\Research\NOAA-NCDC\Temperature\summertemp_1981-2010_geo.dta", clear
> export excel using "E:\Research\NOAA-NCDC\Temperature\summertemp_1981-2010_geo.xls", firstrow(variables)
```

Finally, see final section of Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean summer temperature at the county population-weighted centroids.

## 5. Mean annual relative humidity

Using information from the GSOD data (see Section 3 Mean Winter Temperature), one can calculate the relative humidity.

To calculate the relative humidity, I used the following equation:<sup>3</sup>

---

<sup>3</sup> Insert reference. Talk to Lucia.



$$HR = \frac{\text{vapor pressure}}{\text{saturation pressure}} * 100 = \frac{e_a}{e_s} * 100$$

$$e_a = AT_d^b e^{\left(\frac{-6801.27}{T_d}\right)}$$

$$e_s = AT_s^b e^{\left(\frac{-6801.27}{T_s}\right)}$$

$$HR = \left(\frac{T_d}{T_s}\right)^b e^{\left(\frac{6801.27}{T_s} - \frac{6801.27}{T_d}\right)} * 100$$

Where:

Td: Dew point in Kelvin

Ts: Ambient temperature in Kelvin

b: -5.08

Run the following commands to open the dataset, keep the variables of interest and calculate the relative humidity.

```
> use "E:\Research\NOAA-NCDC\data\GSOD\1981-2010.dta", clear
> keep STN WBAN YEARMODA temp DewP
> gen year = floor(YEARMODA/10000)
> gen month = floor((YEARMODA-year*10000)/100)
> gen day = (YEARMODA-year*10000)-month*100
> drop YEARMODA
> sum
> drop if DewP==9999.9|missing(temp)|missing(DewP)
> replace temp = (temp-32)/9*5+273.15
> replace DewP = (DewP-32)/9*5+273.15
> label variable temp "Temperature in Kelvin"
> label variable DewP "Dew point in Kelvin"
> gen HR = (DewP/temp)^-5.08*exp(6801.27/temp-6801.27/DewP)*100
> label variable HR "Relative humidity in % (approximation)"
```

Now you can collapse the data to obtain means for the variables of interest, by running the following commands:

```
> collapse (mean) HR (count) day , by(STN WBAN year month)
```

In order to keep a dataset with good quality, we need to drop the records that have less than 75% of the recorded period.

```
> drop if(month==2&day<21)
> drop if((month==1|month==3|month==5|month==7|month==8|month==10|month==12)&day<23)
> drop if((month==4|month==6|month==9|month==11)&day<22)
> tab day month
> collapse (mean) HR , by(STN WBAN month)
> collapse (mean) HR (count) month, by(STN WBAN)
> drop if month<9
> drop month
> rename HR annual_HR
> label variable annual_HR "Mean annual relative humidity in % (approximation), 1981-2010"
> save "E:\Research\NOAA-NCDC\Relative_Humidity\annual_HR_1981-2010.dta"
```

I have collapsed the data first by year and month and then by month to obtain monthly averages and then average them over a year to obtain the relative humidity of a typical year, assigning an equal weight to each month. One can run a similar procedure to obtain the July mean (see Section 6).

Once the collapse is performed, you can merge this file with the station location file and drop the observations that are not within the coterminous U.S.

```
> use "E:\Research\NOAA-NCDC\Relative_Humidity\annual_HR_1981-2010.dta", clear
> merge 1:1 STN WBAN using "E:\Research\NOAA-NCDC\station_location\GSOD\ish-history-stations.dta"
> keep if _merge==3
> drop _merge
> sort STN WBAN
> drop if(latitude<23)
> drop if(latitude>50)
> drop if(longitude<-126)
> drop if(longitude>-65)
> save "E:\Research\NOAA-NCDC\Relative_Humidity\annual_HR_1981-2010_geo.dta"
> export excel using "E:\Research\NOAA-NCDC\Relative_Humidity\annual_HR_1981-2010_geo.xls", firstrow(variables)
```

Finally, see final section of Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean summer temperature at the county population-weighted centroids.

## 6. Mean July relative humidity

Using information from the GSOD data (See Mean annual relative humidity, Section 5), one can calculate the relative humidity. To do this, one can run the following commands:

```
> use "E:\Research\NOAA-NCDC\data\GSOD\1981-2010.dta", clear
> keep STN WBAN YEARMODA temp DewP
> gen year = floor(YEARMODA/10000)
> gen month = floor((YEARMODA-year*10000)/100)
> gen day = (YEARMODA-year*10000)-month*100
> keep if month==7
> drop YEARMODA
> sum
> drop if DewP==9999.9|missing(temp)|missing(DewP)
> replace temp = (temp-32)/9*5+273.15
> replace DewP = (DewP-32)/9*5+273.15
> label variable temp "Temperature in Kelvin"
> label variable DewP "Dew point in Kelvin"
> gen HR = (DewP/temp)^-5.08*exp(6801.27/temp-6801.27/DewP)*100
> label variable HR "Relative humidity in % (approximation)"
> collapse (mean) HR (count) day , by(STN WBAN year month)
```

In order to keep a dataset with good quality, we need to drop the records that have less than 75% of the recorded period. Also, since we have previously drop all observations but the ones from July, we need not to drop observations from other months.

```
> drop if day<23
> tab day month
> collapse (mean) HR , by(STN WBAN)
> rename HR July_HR
> label variable July_HR "Mean July relative humidity in % (approximation), 1981-2010"
> save "E:\Research\NOAA-NCDC\Relative_Humidity\July_HR_1981-2010.dta"
```

Then we can merge this file with the station location file.

```
> use "E:\Research\NOAA-NCDC\Relative_Humidity\July_HR_1981-2010.dta", clear
> merge 1:1 STN WBAN using "E:\Research\NOAA-NCDC\station_location\GSOD\ish-history-stations.dta"
> keep if _merge==3
> drop _merge
> sort STN WBAN
> drop if(latitude<23)
> drop if(latitude>50)
> drop if(longitude<-126)
> drop if(longitude>-65)
> save "E:\Research\NOAA-NCDC\Relative_Humidity\July_HR_1981-2010_geo.dta"
> export excel using "E:\Research\NOAA-NCDC\Relative_Humidity\July_HR_1981-2010_geo.xls", firstrow(variables)
```

Finally, see final section of Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean summer temperature at the county population-weighted centroids.

## 7. Mean annual heating degree days (HDD)

### 1981-2010 climate normals

See Mean annual Precipitation (Section 2) for a detailed explanation of where to obtain the 1981-2010 normals dataset and how to process it. I ran the following commands:

```
> insheet using "E:\Research\NOAA-NCDC\data\1981-2010Normals\ann-hydd-normal.txt", clear
> gen length = length(v1)
> gen stationid = substr(v1,1,11)
> gen v2 = substr(v1,12,length)
> drop length v1
> gen length = length(v2)
> sum length
> gen hdd = substr(v2,1,12)
> gen flag = substr(v2,13,.)
> destring hdd , replace
> drop v2 length
> label variable stationid "Station ID"
> label variable hdd "Heating degree days in degrees Fahrenheit, 1981-2010"
> sum hdd
> replace hdd = 0 if hdd == -7777
> sum hdd
> save "E:\Research\NOAA-NCDC\Temperature\hdd_1981-2010.dta"
```

Then we can merge this file with the station location file.

```
> use "E:\Research\NOAA-NCDC\Temperature\hdd_1981-2010.dta", clear
> merge 1:1 stationid using "E:\Research\NOAA-NCDC\station_location\normals\allstations.dta"
> keep if _merge==3
> drop elev name state gsnflag hcnflag WMOID _merge
> drop if missing(hdd)
> label variable latitude "Latitude in decimal degrees"
> label variable longitude "Longitude in decimal degrees"
> drop if(latitude<23)
> drop if(latitude>50)
> drop if(longitude<-126)
> drop if (longitude>-65)
> save "E:\Research\NOAA-NCDC\Temperature\hdd_1981-2010_geo.dta"
> export excel using "E:\Research\NOAA-NCDC\Temperature\hdd_1981-2010_geo.xls", firstrow(variables)
```

Finally, see final section of Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean summer temperature at the county population-weighted centroids.

### 1971-2000 climate normals

See Section 2 Mean annual precipitation to download the 1971-2000 climate normal and import the station location dta file. We need to create a dictionary file in order to easily import the degree-days data into Stata. The dictionary file has the following syntax:

```
degree_days.dct
infile dictionary{
  _column(1) str3 station_number %3s "Station number"
  _column(5) str21 station_name %21s "Station name"
  _column(29) str4 element %4s "CDD or HDD"
  _column(106) str7 degree_days %7s "degree days in degrees Fahrenheit, 1971-2000"
}
```

Run the following commands to generate the degree days dta file:

```
> infile using "E:\Research\NOAA-NCDC\data\1971-2000Normals\degree_days.dct", using("E:\Research\NOAA-NCDC\data\1971-2000Normals\ALnorm.txt") clear
> keep if element=="CDD"|element=="HDD"|element=="CDD*"|element=="HDD*"
> gen cdd = degree_days if(element=="CDD"|element=="CDD*")
> gen hdd = degree_days if(element=="HDD"|element=="HDD*")
> replace station_number = station_number[_n-1] if missing(station_number)
> replace station_name = station_name[_n-1] if missing(station_name)
> destring , replace
> replace cdd = 0 if missing(cdd)
> replace hdd = 0 if missing(hdd)
> collapse (sum) cdd hdd , by(station_number station_name)
> label variable cdd "Cooling degree days in Fahrenheit degrees, 1971-2000"
> label variable hdd "Heating degree days in Fahrenheit degrees, 1971-2000"
> merge 1:1 station_number using "E:\Research\NOAA-NCDC\data\1971-2000Normals\AL_stations.dta"
> drop if _merge==2
> save "E:\Research\NOAA-NCDC\data\1971-2000Normals\AL_degreedays.dta"
```

Then we need to append the precipitation dta files together, and finally import them into ArcGIS.

```
> use "E:\Research\NOAA-NCDC\data\1971-2000Normals\AL_degreedays.dta", clear
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\AR_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\AZ_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\CA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\CO_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\CT_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\DE_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\FL_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\GA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\IA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\ID_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\IL_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\IN_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\KS_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\KY_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\LA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MD_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\ME_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MI_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MN_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MO_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MS_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MT_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NC_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\ND_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NE_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NH_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NJ_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NM_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NV_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NY_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\OH_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\OK_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\OR_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\PA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\RI_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\SC_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\SD_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\TN_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\TX_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\UT_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\VA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\VT_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WI_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WV_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WY_degreedays.dta"
> drop station_number _merge cdd
> order COOPID state station_name hdd , first
> save "E:\Research\NOAA-NCDC\Temperature\hdd_1971-2000_geo.dta"
```

```
> export excel using "E:\Research\NOAA-NCDC\Temperature\hdd_1971-2000_geo.xls", firstrow(variables)
```

Finally, see final section of 1981-2010 climate normals Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean winter temperature at the county population-weighted centroids.

## 8. Mean annual cooling degree days (CDD)

### 1981-2010 climate normals

See Mean annual Precipitation (Section 2) for a detailed explanation of where to obtain the 1981-2010 normals dataset and how to process it. I ran the following commands:

```
> insheet using "E:\Research\NOAA-NCDC\data\1981-2010Normals\ann-cldd-normal.txt", clear
> gen length = length(v1)
> gen stationid = substr(v1,1,11)
> gen v2 = substr(v1,12,length)
> drop length v1
> gen length = length(v2)
> sum length
> gen cdd = substr(v2,1,12)
> gen flag = substr(v2,13,.)
> destring cdd , replace
> drop v2 length
> label variable stationid "Station ID"
> label variable cdd "Cooling degree days in degrees Fahrenheit, 1981-2010"
> sum cdd
> replace cdd = 0 if cdd == -7777
> sum cdd
> save "E:\Research\NOAA-NCDC\Temperature\cdd_1981-2010.dta"
```

Then we can merge this file with the station location file.

```
> use "E:\Research\NOAA-NCDC\Temperature\cdd_1981-2010.dta", clear
> merge 1:1 stationid using "E:\Research\NOAA-NCDC\station_location\normals\allstations.dta"
> keep if _merge==3
> drop elev name state gsnflag hcnflag WMOID _merge
> drop if missing(cdd)
> label variable latitude "Latitude in decimal degrees"
> label variable longitude "Longitude in decimal degrees"
> drop if(latitude<23)
> drop if(latitude>50)
> drop if(longitude<-126)
> drop if (longitude>-65)
> save "E:\Research\NOAA-NCDC\Temperature\cdd_1981-2010_geo.dta"
> export excel using "E:\Research\NOAA-NCDC\Temperature\cdd_1981-2010_geo.xls", firstrow(variables)
```

Finally, see final section of Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean summer temperature at the county population-weighted centroids.

### 1971-2000 climate normals

See Section 2 Mean annual precipitation to download the 1971-2000 climate normals and import the station location dta file. Then see Section 8 Mean annual cooling degree days (CDD) to import the degree days dta file. Since we have already created the degree days dta files by state, we need to append them together and import them into ArcGIS.

```
> use "E:\Research\NOAA-NCDC\data\1971-2000Normals\AL_degreedays.dta", clear
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\AR_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\AZ_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\CA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\CO_degreedays.dta"
```

```
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\CT_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\DE_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\FL_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\GA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\IA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\ID_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\IL_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\IN_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\KS_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\KY_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\LA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MD_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\ME_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MI_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MN_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MO_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MS_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\MT_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NC_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\ND_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NE_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NH_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NJ_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NM_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NV_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\NY_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\OH_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\OK_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\OR_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\PA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\RI_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\SC_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\SD_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\TN_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\TX_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\UT_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\VA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\VT_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WA_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WI_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WV_degreedays.dta"
> append using "E:\Research\NOAA-NCDC\data\1971-2000Normals\WY_degreedays.dta"
> drop station_number _merge hdd
> order COOPID state station_name cdd , first
> save "E:\Research\NOAA-NCDC\Temperature\cdd_1971-2000_geo.dta"
> export excel using "E:\Research\NOAA-NCDC\Temperature\cdd_1971-2000_geo.xls", firstrow(variables)
```

Finally, see final section of 1981-2010 climate normals Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean winter temperature at the county population-weighted centroids.

## 9. Mean annual wind speed

Using information from the GSOD data (See Mean Winter Temperature, Section 3), one can calculate the mean annual wind speed. Remember that the wind speed is in knots, so we need to convert it to m.p.h. somewhere To do this, one can run the following commands:

```
> use "E:\Research\NOAA-NCDC\data\GSOD\1981-2010.dta", clear
> keep STN WBAN YEARMODA WDSP
> gen year = floor(YEARMODA/10000)
> gen month = floor((YEARMODA-year*10000)/100)
> gen day = (YEARMODA-year*10000)-month*100
> drop YEARMODA
> drop if missing(WDSP) | WDSP==999.9
```

Then, we can collapse the information in order to obtain the mean temperature during this period.

```
> collapse (mean) WDSP (count) day , by (STN WBAN year month)
> drop if (month==2&day<21)
> drop if ( (month==1|month==3|month==5|month==7|month==8|month==10|month==12) &day<23)
> drop if ( (month==4|month==6|month==9|month==11) &day<22)
> tab day month
> collapse (mean) WDSP , by (STN WBAN month)
> collapse (mean) WDSP (count) month, by (STN WBAN)
> drop if month<9
> drop month
> rename WDSP wind
> replace wind = wind/0.868976
> label variable wind "Mean annual wind speed in mph, 1981-2010"
> sort STN WBAN
> save "E:\Research\NOAA-NCDC\Wind_Speed\wind_1981-2010.dta" , replace
> merge 1:1 STN WBAN using "E:\Research\NOAA-NCDC\station_location\GSOD\ish-history-stations.dta"
> keep if _merge==3
> drop _merge
> sort STN WBAN
> drop if (latitude<23)
> drop if (latitude>50)
> drop if (longitude<-126)
> drop if (longitude>-65)
> save "E:\Research\NOAA-NCDC\Wind_Speed\wind_1981-2010_geo.dta"
> export excel using "E:\Research\NOAA-NCDC\Wind_Speed\wind_1981-2010_geo.xls", firstrow(variables)
```

Finally, see final section of Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean summer temperature at the county population-weighted centroids.

## 10. Mean annual sunshine

Getting sunshine data proved to be relatively more difficult. Contact the NCDC staff if you have trouble finding this data ([ncdc.orders@noaa.gov](mailto:ncdc.orders@noaa.gov)). They provided the following ftp site:  
<ftp://ftp3.ncdc.noaa.gov/pub/data/3210>

This ftp site has daily meteorological information from various stations. It is very important to read the readme.txt file (<ftp://ftp3.ncdc.noaa.gov/pub/data/3210/README.TXT>) and also download the data set documentation (<http://www1.ncdc.noaa.gov/pub/data/documentlibrary/tddoc/td3210.doc>) in order to create the appropriate dictionary file.

Also, the files are available per month so first one needs to merge these files into a single file per year. It is easier to accomplish this using the command prompt. You need to open the "Command Prompt" program, and then change the directory until you get to the folder in which you have the files you want to merge. For instance, the initial directory may be "C:\Users\diegohornamunoz" and you stored the files in the directory "E:\Research\NOAA-NCDC\data\3210\1981"

In order to move to this subfolder in a different drive you need to type:

```
> E:
> cd Research
> cd NOAA-NCDC
> cd data
> cd 3210
> cd by_year
> cd 1981
```

To move up to a subfolder you need to type:

```
> cd..
```

Then once you are in the directory where the files are stored, you need to type:

```
> Copy *. 1981.txt
```

By doing this, you will have created a new .txt file which contains all the information from the files in that folder.

The dictionary file used to import the merge file into Stata has the following syntax:

```
infile dictionary{
_column(1) str3 rectype %3s "Record type"
_column(4) str8 ID %8s "WBAN Station ID"
_column(12) str4 element %4s "Meteorological element type"
_column(16) str2 unit %2s "Meteorological element measurement units code"
_column(18) str4 year %4s "Year"
_column(22) str2 month %2s "Month"
_column(24) str1 srccode1 %1s "Source Code 1"
_column(25) str1 srccode2 %1s "Source Code 2"
_column(26) str2 future %2s "Reserved for future use"
_column(28) str3 groups %3s "Number of Data Portion Groups That Follow"
_column(31) str2 day1 %2s "Day of month"
_column(33) str2 hour1 %2s "Hour of observation"
_column(35) str1 sign1 %1s "Sign of meteorological value"
_column(36) str5 value1 %5s "Value of meteorological value"
_column(41) str1 Flag1_1 %1s "Quality Control Flag 1"
_column(42) str1 Flag2_1 %1s "Quality Control Flag 2"
_column(43) str2 day2 %2s "Day of month"
_column(45) str2 hour2 %2s "Hour of observation"
_column(47) str1 sign2 %1s "Sign of meteorological value"
_column(48) str5 value2 %5s "Value of meteorological value"
_column(53) str1 Flag1_2 %1s "Quality Control Flag 1"
_column(54) str1 Flag2_2 %1s "Quality Control Flag 2"
_column(55) str2 day3 %2s "Day of month"
_column(57) str2 hour3 %2s "Hour of observation"
_column(59) str1 sign3 %1s "Sign of meteorological value"
_column(60) str5 value3 %5s "Value of meteorological value"
_column(65) str1 Flag1_3 %1s "Quality Control Flag 1"
_column(66) str1 Flag2_3 %1s "Quality Control Flag 2"
_column(67) str2 day4 %2s "Day of month"
_column(69) str2 hour4 %2s "Hour of observation"
_column(71) str1 sign4 %1s "Sign of meteorological value"
_column(72) str5 value4 %5s "Value of meteorological value"
_column(77) str1 Flag1_4 %1s "Quality Control Flag 1"
_column(78) str1 Flag2_4 %1s "Quality Control Flag 2"
_column(79) str2 day5 %2s "Day of month"
_column(81) str2 hour5 %2s "Hour of observation"
_column(83) str1 sign5 %1s "Sign of meteorological value"
_column(84) str5 value5 %5s "Value of meteorological value"
_column(89) str1 Flag1_5 %1s "Quality Control Flag 1"
_column(90) str1 Flag2_5 %1s "Quality Control Flag 2"
_column(91) str2 day6 %2s "Day of month"
_column(93) str2 hour6 %2s "Hour of observation"
_column(95) str1 sign6 %1s "Sign of meteorological value"
_column(96) str5 value6 %5s "Value of meteorological value"
_column(101) str1 Flag1_6 %1s "Quality Control Flag 1"
_column(102) str1 Flag2_6 %1s "Quality Control Flag 2"
_column(103) str2 day7 %2s "Day of month"
_column(105) str2 hour7 %2s "Hour of observation"
_column(107) str1 sign7 %1s "Sign of meteorological value"
_column(108) str5 value7 %5s "Value of meteorological value"
_column(113) str1 Flag1_7 %1s "Quality Control Flag 1"
_column(114) str1 Flag2_7 %1s "Quality Control Flag 2"
_column(115) str2 day8 %2s "Day of month"
_column(117) str2 hour8 %2s "Hour of observation"
_column(119) str1 sign8 %1s "Sign of meteorological value"
_column(120) str5 value8 %5s "Value of meteorological value"
_column(125) str1 Flag1_8 %1s "Quality Control Flag 1"
```



```

_column(126) str1 Flag2_8 %1s "Quality Control Flag 2"
_column(127) str2 day9 %2s "Day of month"
_column(129) str2 hour9 %2s "Hour of observation"
_column(131) str1 sign9 %1s "Sign of meteorological value"
_column(132) str5 value9 %5s "Value of meteorological value"
_column(137) str1 Flag1_9 %1s "Quality Control Flag 1"
_column(138) str1 Flag2_9 %1s "Quality Control Flag 2"
_column(139) str2 day10 %2s "Day of month"
_column(141) str2 hour10 %2s "Hour of observation"
_column(143) str1 sign10 %1s "Sign of meteorological value"
_column(144) str5 value10 %5s "Value of meteorological value"
_column(149) str1 Flag1_10 %1s "Quality Control Flag 1"
_column(150) str1 Flag2_10 %1s "Quality Control Flag 2"
_column(151) str2 day11 %2s "Day of month"
_column(153) str2 hour11 %2s "Hour of observation"
_column(155) str1 sign11 %1s "Sign of meteorological value"
_column(156) str5 value11 %5s "Value of meteorological value"
_column(161) str1 Flag1_11 %1s "Quality Control Flag 1"
_column(162) str1 Flag2_11 %1s "Quality Control Flag 2"
_column(163) str2 day12 %2s "Day of month"
_column(165) str2 hour12 %2s "Hour of observation"
_column(167) str1 sign12 %1s "Sign of meteorological value"
_column(168) str5 value12 %5s "Value of meteorological value"
_column(173) str1 Flag1_12 %1s "Quality Control Flag 1"
_column(174) str1 Flag2_12 %1s "Quality Control Flag 2"
_column(175) str2 day13 %2s "Day of month"
_column(177) str2 hour13 %2s "Hour of observation"
_column(179) str1 sign13 %1s "Sign of meteorological value"
_column(180) str5 value13 %5s "Value of meteorological value"
_column(185) str1 Flag1_13 %1s "Quality Control Flag 1"
_column(186) str1 Flag2_13 %1s "Quality Control Flag 2"
_column(187) str2 day14 %2s "Day of month"
_column(189) str2 hour14 %2s "Hour of observation"
_column(191) str1 sign14 %1s "Sign of meteorological value"
_column(192) str5 value14 %5s "Value of meteorological value"
_column(197) str1 Flag1_14 %1s "Quality Control Flag 1"
_column(198) str1 Flag2_14 %1s "Quality Control Flag 2"
_column(199) str2 day15 %2s "Day of month"
_column(201) str2 hour15 %2s "Hour of observation"
_column(203) str1 sign15 %1s "Sign of meteorological value"
_column(204) str5 value15 %5s "Value of meteorological value"
_column(209) str1 Flag1_15 %1s "Quality Control Flag 1"
_column(210) str1 Flag2_15 %1s "Quality Control Flag 2"
_column(211) str2 day16 %2s "Day of month"
_column(213) str2 hour16 %2s "Hour of observation"
_column(215) str1 sign16 %1s "Sign of meteorological value"
_column(216) str5 value16 %5s "Value of meteorological value"
_column(221) str1 Flag1_16 %1s "Quality Control Flag 1"
_column(222) str1 Flag2_16 %1s "Quality Control Flag 2"
_column(223) str2 day17 %2s "Day of month"
_column(225) str2 hour17 %2s "Hour of observation"
_column(227) str1 sign17 %1s "Sign of meteorological value"
_column(228) str5 value17 %5s "Value of meteorological value"
_column(233) str1 Flag1_17 %1s "Quality Control Flag 1"
_column(234) str1 Flag2_17 %1s "Quality Control Flag 2"
_column(235) str2 day18 %2s "Day of month"
_column(237) str2 hour18 %2s "Hour of observation"
_column(239) str1 sign18 %1s "Sign of meteorological value"
_column(240) str5 value18 %5s "Value of meteorological value"
_column(245) str1 Flag1_18 %1s "Quality Control Flag 1"
_column(246) str1 Flag2_18 %1s "Quality Control Flag 2"
_column(247) str2 day19 %2s "Day of month"
_column(249) str2 hour19 %2s "Hour of observation"
_column(251) str1 sign19 %1s "Sign of meteorological value"
_column(252) str5 value19 %5s "Value of meteorological value"
_column(257) str1 Flag1_19 %1s "Quality Control Flag 1"
_column(258) str1 Flag2_19 %1s "Quality Control Flag 2"
_column(259) str2 day20 %2s "Day of month"
_column(261) str2 hour20 %2s "Hour of observation"
_column(263) str1 sign20 %1s "Sign of meteorological value"
_column(264) str5 value20 %5s "Value of meteorological value"
_column(269) str1 Flag1_20 %1s "Quality Control Flag 1"
_column(270) str1 Flag2_20 %1s "Quality Control Flag 2"
_column(271) str2 day21 %2s "Day of month"

```

```

_column(273) str2 hour21 %2s "Hour of observation"
_column(275) str1 sign21 %1s "Sign of meteorological value"
_column(276) str5 value21 %5s "Value of meteorological value"
_column(281) str1 Flag1_21 %1s "Quality Control Flag 1"
_column(282) str1 Flag2_21 %1s "Quality Control Flag 2"
_column(283) str2 day22 %2s "Day of month"
_column(285) str2 hour22 %2s "Hour of observation"
_column(287) str1 sign22 %1s "Sign of meteorological value"
_column(288) str5 value22 %5s "Value of meteorological value"
_column(293) str1 Flag1_22 %1s "Quality Control Flag 1"
_column(294) str1 Flag2_22 %1s "Quality Control Flag 2"
_column(295) str2 day23 %2s "Day of month"
_column(297) str2 hour23 %2s "Hour of observation"
_column(299) str1 sign23 %1s "Sign of meteorological value"
_column(300) str5 value23 %5s "Value of meteorological value"
_column(305) str1 Flag1_23 %1s "Quality Control Flag 1"
_column(306) str1 Flag2_23 %1s "Quality Control Flag 2"
_column(307) str2 day24 %2s "Day of month"
_column(309) str2 hour24 %2s "Hour of observation"
_column(311) str1 sign24 %1s "Sign of meteorological value"
_column(312) str5 value24 %5s "Value of meteorological value"
_column(317) str1 Flag1_24 %1s "Quality Control Flag 1"
_column(318) str1 Flag2_24 %1s "Quality Control Flag 2"
_column(319) str2 day25 %2s "Day of month"
_column(321) str2 hour25 %2s "Hour of observation"
_column(323) str1 sign25 %1s "Sign of meteorological value"
_column(324) str5 value25 %5s "Value of meteorological value"
_column(329) str1 Flag1_25 %1s "Quality Control Flag 1"
_column(330) str1 Flag2_25 %1s "Quality Control Flag 2"
_column(331) str2 day26 %2s "Day of month"
_column(333) str2 hour26 %2s "Hour of observation"
_column(335) str1 sign26 %1s "Sign of meteorological value"
_column(336) str5 value26 %5s "Value of meteorological value"
_column(341) str1 Flag1_26 %1s "Quality Control Flag 1"
_column(342) str1 Flag2_26 %1s "Quality Control Flag 2"
_column(343) str2 day27 %2s "Day of month"
_column(345) str2 hour27 %2s "Hour of observation"
_column(347) str1 sign27 %1s "Sign of meteorological value"
_column(348) str5 value27 %5s "Value of meteorological value"
_column(353) str1 Flag1_27 %1s "Quality Control Flag 1"
_column(354) str1 Flag2_27 %1s "Quality Control Flag 2"
_column(355) str2 day28 %2s "Day of month"
_column(357) str2 hour28 %2s "Hour of observation"
_column(359) str1 sign28 %1s "Sign of meteorological value"
_column(360) str5 value28 %5s "Value of meteorological value"
_column(365) str1 Flag1_28 %1s "Quality Control Flag 1"
_column(366) str1 Flag2_28 %1s "Quality Control Flag 2"
_column(367) str2 day29 %2s "Day of month"
_column(369) str2 hour29 %2s "Hour of observation"
_column(371) str1 sign29 %1s "Sign of meteorological value"
_column(372) str5 value29 %5s "Value of meteorological value"
_column(377) str1 Flag1_29 %1s "Quality Control Flag 1"
_column(378) str1 Flag2_29 %1s "Quality Control Flag 2"
_column(379) str2 day30 %2s "Day of month"
_column(381) str2 hour30 %2s "Hour of observation"
_column(383) str1 sign30 %1s "Sign of meteorological value"
_column(384) str5 value30 %5s "Value of meteorological value"
_column(389) str1 Flag1_30 %1s "Quality Control Flag 1"
_column(390) str1 Flag2_30 %1s "Quality Control Flag 2"
_column(391) str2 day31 %2s "Day of month"
_column(393) str2 hour31 %2s "Hour of observation"
_column(395) str1 sign31 %1s "Sign of meteorological value"
_column(396) str5 value31 %5s "Value of meteorological value"
_column(401) str1 Flag1_31 %1s "Quality Control Flag 1"
_column(402) str1 Flag2_31 %1s "Quality Control Flag 2"
_column(403) str100 rest
}

```

Once the text file has been imported into Stata, you need to keep the sunshine records and then transform the dataset from 'wide' to 'long'. 'Wide' means that records are stored in columns rather than rows; this makes the dataset hard to work with, so it is very important to transpose the data. Run the following commands:

```
> infile using "E:\Research\NOAA-NCDC\data\3210\metday_data_3210.dct", using("E:\Research\NOAA-NCDC\data\3210\1981.txt") clear
> keep if element=="PSUN"
> reshape long value Flag1_ Flag2_, i(ID element year month) j(day 1-31) string
> keep ID element year month day value Flag1_ Flag2_
> destring , replace
> sort ID year month day
```

After doing this, you will end up with a dataset that has only the following variables:

|         |                             |
|---------|-----------------------------|
| ID      | WBAN Station ID             |
| element | Meteorological element type |
| year    | Year                        |
| month   | Month                       |
| day     |                             |
| value   |                             |
| Flag1_  |                             |
| Flag2_  |                             |

Now, it is also important to drop the values that have some sort of suspicious flags. The following tables describe the variables Flag1\_ and Flag2\_.

| Flag1_ value | Meaning  |
|--------------|--|
| A            | Accumulated amount. This value is the amount accumulated since the last measurement. (SNOW, SNWD, PRCP)  |
| B            | Accumulated amount. Value includes estimated values. (SNOW, SNWD, PRCP)  |
| D            | Derived value.   |
| E            | Estimated value.   |
| M            | Data Element Missing. This is for fixed length records only.   |
| P            | Multiple-occurrence Peak Gust. Last occurrence is indicated. (PKGS , FSIN , FSMI)  |
| S            | Included in a Subsequent Value. This precipitation amount is being accumulated. Total will be included in a subsequent value. (TPCP, SNOW, SNWD) |
| T            | Trace of Precipitation, Snowfall or Snow depth. Value would be '00000'. (TPCP, SNOW, SNWD)   |
| b            | (blank) Not needed.  |

| Flag2_ value | Meaning   |
|--------------|---|
| 0            | Observed data has passed all internal consistency checks.   |
| 1            | Validity indeterminable (primarily for pre-1984 data)   |
| 2            | Observed data has failed an internal consistency check - (subsequent edited value follows observed value)<br>Data prior to 1 January 1984 = Observed data exceeded preselected climatological limits during conversion from historic TD-9750 files. (No edited value follows)   |
| 3            | Data beginning 1 January 1984 through 1988 and data beginning 1996 through current = Observed data has failed an internal consistency check. (No edited value follows) (Low level of confidence of observed value)<br><br>Data beginning 1989 through 1995 = Observed data has failed an internal consistency check but passed a manual inspection of the data. (No edited value follows) |

|   |   |
|---|---|
| 4 | Observed data value invalid. (No edited value follows)  |
| 5 | Data converted from historic TD-9750 files exceeded all known climatological extremes. No edited value follows)   |
| A | Observed data has failed an internal consistency check but passed a manual inspection of the data. (No edited value follows) (High level of confidence of observed value) |
| D | Wind direction code is invalid (PKGS through December 1983 only)  |
| E | Edited data value passes all systems checks - no observed value present   |
| S | Manually edited value passes all systems checks.  |

Thus, we need to run the tab command (`tab Flag1_`) and assess if we need to drop any observation or not, then we can drop the flag variables since they are no longer needed.

Then, once we have an independent file per year we can append these files and have one single file for the 1981-2010 period. Then we can run the following commands for collapsing the information and drop the records that have less than 75% of the recorded period.

```
> use "E:\Research\NOAA-NCDC\data\3210\1981.dta", clear
> append using "E:\Research\NOAA-NCDC\data\3210\1982.dta" "E:\Research\NOAA-NCDC\data\3210\1983.dta"
"E:\Research\NOAA-NCDC\data\3210\1984.dta" "E:\Research\NOAA-NCDC\data\3210\1985.dta" "E:\Research\NOAA-
NCDC\data\3210\1986.dta" "E:\Research\NOAA-NCDC\data\3210\1987.dta" "E:\Research\NOAA-NCDC\data\3210\1988.dta"
"E:\Research\NOAA-NCDC\data\3210\1989.dta" "E:\Research\NOAA-NCDC\data\3210\1990.dta" "E:\Research\NOAA-
NCDC\data\3210\1991.dta" "E:\Research\NOAA-NCDC\data\3210\1992.dta" "E:\Research\NOAA-NCDC\data\3210\1993.dta"
"E:\Research\NOAA-NCDC\data\3210\1994.dta" "E:\Research\NOAA-NCDC\data\3210\1995.dta" "E:\Research\NOAA-
NCDC\data\3210\1996.dta" "E:\Research\NOAA-NCDC\data\3210\1997.dta" "E:\Research\NOAA-NCDC\data\3210\1998.dta"
"E:\Research\NOAA-NCDC\data\3210\1999.dta" "E:\Research\NOAA-NCDC\data\3210\2000.dta" "E:\Research\NOAA-
NCDC\data\3210\2001.dta" "E:\Research\NOAA-NCDC\data\3210\2002.dta" "E:\Research\NOAA-NCDC\data\3210\2003.dta"
"E:\Research\NOAA-NCDC\data\3210\2004.dta" "E:\Research\NOAA-NCDC\data\3210\2005.dta" "E:\Research\NOAA-
NCDC\data\3210\2006.dta" "E:\Research\NOAA-NCDC\data\3210\2007.dta" "E:\Research\NOAA-NCDC\data\3210\2008.dta"
"E:\Research\NOAA-NCDC\data\3210\2009.dta" "E:\Research\NOAA-NCDC\data\3210\2010.dta"
> sort ID year month day
> rename value sunshine
> label variable sunshine "Percent of possible sunshine"
> label variable day "Day"
> drop if sunshine==99999|missing(sunshine)
> collapse (mean) sunshine (count) day , by(ID year month)
> drop if month==2&day<21
> drop if (month==1|month==3|month==5|month==7|month==8|month==10|month==12)&day<23)
> drop if (month==4|month==6|month==9|month==11)&day<22)
> tab day month
> collapse sunshine , by(ID month)
> collapse sunshine (count) month , by(ID)
> drop if month<9
> drop month
> label variable sunshine "Annual mean percent of possible sunshine, 1981-2010"
> rename ID WBAN
> save "E:\Research\NOAA-NCDC\Sunshine\sunshine_1981-2010.dta"
```

Now that we have collapsed the information, we need to merge it with station location information. We can use the same station location dataset used for GSOD datasets. Once the files are merged, you might notice that you needed to select a one-to-many cardinality, this is because the location dataset has two codes that describe the same station and in some instances, it even has multiple codes for the same climate station. Therefore, you might have duplicate or multiple records for the same observation. I ran the following commands to get rid of the stations that were outside the contiguous U.S. and of the duplicate records.

```
> use "E:\Research\NOAA-NCDC\Sunshine\sunshine_1981-2010.dta", clear
> merge 1:m WBAN using "E:\Research\NOAA-NCDC\station_location\GSOD\ish-history-stations.dta"
> keep if _merge==3
> drop _merge
> drop if (latitude<23)|(latitude>50)|(longitude<-126)|(longitude>-65)
> gen sunshine2 = sunshine
> collapse (min) sunshine (max) sunshine2 , by(WBAN latitude longitude)
```

```
> gen diff = sunshine-sunshine2
> sum diff
```

If the difference between sunshine and sunshine2 is really close to zero, then this means that the observations were in fact duplicate and that no different value was assigned to a station located on top of another station. Then I ran the following commands to obtain the final clean dataset.

```
> drop sunshine2 diff
> label variable sunshine "Annual mean percent of possible sunshine, 1981-2010"
> save "E:\Research\NOAA-NCDC\Sunshine\sunshine_1981-2010_geo.dta"
> export excel using "E:\Research\NOAA-NCDC\Sunshine\sunshine_1981-2010_geo.xls", firstrow(variables)
```

Finally, see final section of Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean summer temperature at the county population-weighted centroids.

## 11. Heavy fog

Using information from the GSOD data (see Section 3 Mean Winter Temperature), one can calculate the heavy fog variable. The heavy fog unit is 'number of days with visibility equal to or less than 0.25 miles'. I ran the following commands:

```
> use "E:\Research\NOAA-NCDC\data\GSOD\1981-2010_visib.dta" , clear
> drop if missing(visib)|visib==999.9
> label variable visib "visibility in miles"
```

Since there may be duplicate info in stations that share the same location, we need to merge this dataset with the station location dataset. Also we need to drop the observations that were based on '0' observations (vis\_code=0).

```
> drop if vis_code == 0
> merge m:1 STN WBAN using "E:\Research\NOAA-NCDC\station_location\GSOD\ish-history-stations.dta"
> keep if _merge==3
> drop if(latitude<23)
> drop if(latitude>50)
> drop if(longitude<-126)
> drop if(longitude>-65)
> drop vis_code
> drop _merge
```

We also need to collapse the information in order to get rid of the duplicate information.

```
> collapse (mean) visib , by(longitude latitude YEARMODA)
> outsheet using "E:\Research\NOAA-NCDC\Heavy_Fog\1981-2010_visib.csv", comma replace
```

Once we have a 'csv' file that contains all the dataset that we need, we can import this into R, then keep specific dates and run the spatial interpolation per day to obtain the *estimated* visibility in each county population-weighted centroid. By doing this we are estimating the variable that we expect to vary spatially and not a count variable that may be more difficult to accurately interpolate. See uploaded scripts to M+Box in order to view the code that was used to run the spatial interpolation. Once we have run the spatial interpolation, we can then import the average number of days with visibility equal to or less than 0.25 miles at each one of the county population-weighted centroids.

```
> insheet using "E:\Research\R\visib_19812010.txt", delimiter(" ") clear
> rename fog visibility
> label variable visibility "Mean number of days per annum with visibility <= 0.25 miles"
```

```
> rename fips FIPS
> save "E:\Research\NOAA-NCDC\Heavy_Fog\visib_1981-2010_county.dta"
> use "E:\Research\Research.dta", clear
> merge 1:1 FIPS using "E:\Research\NOAA-NCDC\Heavy_Fog\visib_1981-2010_county.dta"
> drop _merge
> order visibility, after(sunshine)
> save "E:\Research\Research.dta", replace
```

## 12. Percent Water Area

Since we are calculating the 'Percent Water Area', we need to calculate, first, the total land area. We can do this by downloading the county boundary files from the following census website:

<ftp://ftp2.census.gov/geo/tiger/TIGER2010/COUNTY/2010/>. Then we need to download the water area shapefiles from the census website: <ftp://ftp2.census.gov/geo/tiger/TIGER2010/AREAWATER/>.

Once these shapefiles have been downloaded, we need to load them into ArcGIS. You will notice that the shapefiles are in GCS NAD83. This is ok but we will need to project it into an adequate projected coordinate system in order to calculate the areas, and then the percentage. We need to do the following:

- (1) Load the county shapefiles.
- (2) Select the counties that are within the Coterminous U.S.
- (3) Right click the layer and select "Data \ Export data ..."
  - a. Export: Selected features
  - b. Use the same coordinate system as: this layer's source data
  - c. Output feature class: E:\Research\GIS\Shapefiles\County\NAD83\CountyBoundary\_2010.shp
- (4) Project this shapefile. Run the Project tool (Arc Toolbox \ Projections and Transformations \ Feature \ Project)
  - a. Input Dataset or Feature Class: CountyBoundary\_2010
  - b. Output Dataset or Feature Class:  
E:\Research\GIS\Shapefiles\County\EqualArea\CountyBoundary\_2010.shp
  - c. Output Coordinate System: USA Contiguous Albers Equal Area Conic
- (5) Since there is one shapefile per county, we need to merge them together before loading them into ArcMap. Run the Merge tool (Arc Toolbox \ Data Management Tools \ General \ Merge). We need to do this twice, first to merge the water features within each state and then to merge the water features within the Coterminous U.S.
- (6) Load the merged shapefile.
- (7) Project this shapefile. Run the Project tool (Arc Toolbox \ Projections and Transformations \ Feature \ Project).
  - a. Input Dataset or Feature Class: AreaWater\_2010
  - b. Output Dataset or Feature Class:  
E:\Research\GIS\Shapefiles\Water\EqualArea\AreaWater\_2010.shp
  - c. Output Coordinate System: USA Contiguous Albers Equal Area Conic
- (8) Then, since we need the land area of each county, we need to erase the water area out of the county boundary shapefile. Run the Erase tool (Arc Toolbox \ Analysis Tools \ Overlay \ Erase).
  - a. Input Features: CountyBoundary\_2010
  - b. Erase Features: AreaWater\_2010
  - c. Output Feature Class: E:\Research\GIS\Shapefiles\County\EqualArea\CountyLand\_2010.shp
- (9) Open the CountyLand\_2010.shp attribute table.
- (10) Add Field...:
  - a. Name: land\_mi2

- b. Type: Float
- (11) Then right click this new field. Select 'Calculate geometry'
  - a. Property: Area
  - b. Coordinate system: Use coordinate system of the data source: PCS: USA Contiguous Albers Equal Area Conic
  - c. Units: Square Miles US [sq mi]
- (12) Then export the attribute table as a txt file.
- (13) Repeat for the AreaWater\_2010.shp file. Add Field (name: water\_mi2), calculate geometry and export the attribute table.

Now that we have both attribute tables we can import these txt files into Stata, merge them together and calculate the 'percent water area'. Run to the following commands to do so:

```
> insheet using "E:\Research\Census-TIGER\CountyWater_2010.txt"
> gen FIPS = statefp*1000+countyfp
> collapse (sum) water_mi2 , by(FIPS)
> label variable water_mi2 "Water area in square miles"
> save "E:\Research\Census-TIGER\CountyWater_2010.dta"
> insheet using "E:\Research\Census-TIGER\CountyLand_2010.txt" , clear
> gen FIPS = statefp*1000+countyfp
> collapse (sum) land_mi2 , by(FIPS)
> label variable land_mi2 "Land area in square miles"
> save "E:\Research\Census-TIGER\CountyLand_2010.dta"
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyWater_2010.dta"
> drop _merge
> gen PctWater = water_mi2/land_mi2*100
> label variable PctWater "Water area per land area, in percent"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> save "E:\Research\Census-TIGER\PctWater_2010.dta"
```

### 13. Coastal

Two types of coastal information need to be collected.

|                                |                    |
|--------------------------------|--------------------|
| Coast                          | =1 if on coast     |
| Non-adjacent coastal watershed | =2 if in watershed |

The source for 2000 is said to be the Strategic Environmental Assessments Division of the National Oceanic and Atmospheric Administration (NOAA-SEAD). However, this department is not found in the official website (see <http://www.noaa.gov/organizations.html>).

With help from staff at the National Ocean Economic Program (NOEP), coastal information was received in two pdf files: [C:\Data\Coastal\ CZM\\_counties.pdf](#), which is a list of coastal counties by state, and [C:\Data\Coastal\ Watershed\\_Counties.pdf](#), which is a combination of coastal counties non-adjacent coastal watershed counties. Since Professor Bieri thinks the coastal characteristics changes little within a decade, counties in State Michigan and Virginia were tested by comparing 2000 data and new pdf data (see [C:\Data\Coastal\ Coastal\\_info\\_comparison\\_20120308.xlsx](#)). No changes have been found. Therefore, the 2000 data about coastal counties are used for 2010. Data are stored in [C:\Data\Coastal\coast\\_2010.dta](#)

### 14. Mountain peaks

Data of Mountain peaks above 1500 meters can be found in Esri

<http://www.arcgis.com/home/item.html?id=6706f7e6712b4b479dcb4fce4b7b3172>. The single layer can be

added into ArcGIS. The attribute table includes information of FIPS code. Convert the file into STATA, and save as `C:\Data\Mountain\mountainpeak1500.dta`.

## 15. Rivers

Linear hydrographic information can be downloaded from the U.S. Census Bureau website (<http://www.census.gov/geo/maps-data/data/tiger-line.html>). Since the variable is expressed in miles (of river) per square mile (of land), we only need hydrographic data, not hydrological data. However, it is far easier to download the data from the following ftp site: <ftp://ftp2.census.gov/geo/tiger/TIGER2010/LINEARWATER/>.

However, these shapefiles contain river, streams, creeks and canals data; therefore, it contains far more information than we need. Another shapefile was found on the National Atlas website (<http://nationalatlas.gov/atlasftp-na.html?openChapters=chpwater#chpwater>) (<http://nationalatlas.gov/mld/hydro0m.html>). Once we have downloaded the shapefile, we need to project it into GCS NAD83 (ArcToolbox \ Data Management Tools \ Projections and Transformations \ Feature \ Project), and then clip it (ArcToolbox \ Analysis tools \ Extract \ Clip) with the county boundary shapefile in order to keep only the hydrographic information that is within the Contiguous U.S. Then, we need to select the river information. By looking at the attribute table, we can identify that the field records with the field 'Type' equal to 17 and 18 are rivers. Then by selecting by attributes the records that have Type equal to 17 or 18, we can select them and then export the selected features as a shapefile.

After doing this, we need to project this shapefile into a PCS which minimizes distance distortion (i.e. USA Contiguous Equidistant Conic). Once we have projected the shapefile into a PCS, we need to intersect the river and county boundary shapefiles (ArcToolbox \ Analysis Tools \ Overlay \ Intersect). Then, after having intersected both shapefiles we need to calculate the length of the river segments and then calculate the length of each one of them (1. add a field, 2. calculate geometry) and export the attribute table as a txt file. Then, we can import this txt file into Stata and then calculate the river length in miles at the county level and the river length per land area in mile/mile<sup>2</sup>. Run the following commands to accomplish that:

```
> insheet using "E:\Research\NationalAtlas\Hydrography\river_county.txt"
> gen FIPS = state*1000+county
> keep FIPS length_mil
> order FIPS , first
> rename length_mil river_length
> collapse (sum) river_length , by(FIPS)
> label variable river_length "River length in miles"
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyLand_2010.dta"
> sort FIPS
> replace river_length = 0 if missing(river_length)
> drop _merge
> gen river_length_perland = river_length/land_mi2
> drop land_mi2
> label variable river_length_perland "River length per land area in mile/mile^2"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> save "E:\Research\NationalAtlas\Hydrography\River_2010.dta"
```

## 16. Federal land

The latest data is in 2005. It can be accessed in <http://www.nationalatlas.gov/atlasftp.html?openChapters=chpagri%2Cchpgeo1%2Cchpbound#chpbound>. We need to the following to import to obtain the extension of Federal Land in percentage.



- (1) Add the downloaded layer into ArcMap. Project it as GCS NAD83 (ArcToolbox \ Data Management Tools \ Projections and Transformations \ Feature \ Project).
- (2) Then, project it into USA Contiguous Albers Equal Area Conic. We are using this projected coordinate system because we are going to calculate areas; therefore we need to keep areas with minimal distortion (ArcToolbox \ Data Management Tools \ Projections and Transformations \ Feature \ Project).
- (3) Since this shapefile contains the federal area, we need to intersect it with the county\_land shapefile in order to obtain the federal land (i.e., the federal area which is on land). Run the intersect tool (ArcToolbox \ Analysis Tools \ Overlay \ Intersect). Then select the following options:
  - a. Input features: Dissolved Federal area shapefile & county land shapefile.
  - b. Output feature class: Path and name of the new shapefile. This shapefile will have just the federal land.
- (4) Then, we need to open the attribute tables of the newly created shapefile.
- (5) Add a new field named: 'fed\_mile2' and then right click on it and select calculate geometry. Calculate the area in square miles.
- (6) Export the attribute table as a txt file.

Then we can import this table into Stata and use the information to calculate the federal land as percentage of the total land of the county. Run the following commands to import these txt files into Stata and format the information:

```
> insheet using "E:\Research\NationalAtlas\FedLand\data\county_fedarea.txt", clear
> gen FIPS = statefp10*1000+countyfp10
> collapse (sum) fed_mi2 , by(FIPS)
> rename fed_mi2 fedland_mi2
> label variable fedland_mi2 "Federal land in square miles"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> sort FIPS
> drop _merge statecode
> replace fedland_mi2 = 0 if missing(fedland_mi2)
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyLand_2010.dta"
> drop _merge
> gen fedland_pct = fedland_mi2/land_mi2*100
> label variable fedland_pct "Federal land in percentage (over land area)"
> order FIPS state county , first
> drop land_mi2
> save "E:\Research\NationalAtlas\FedLand\FedLand_2010.dta", replace
```

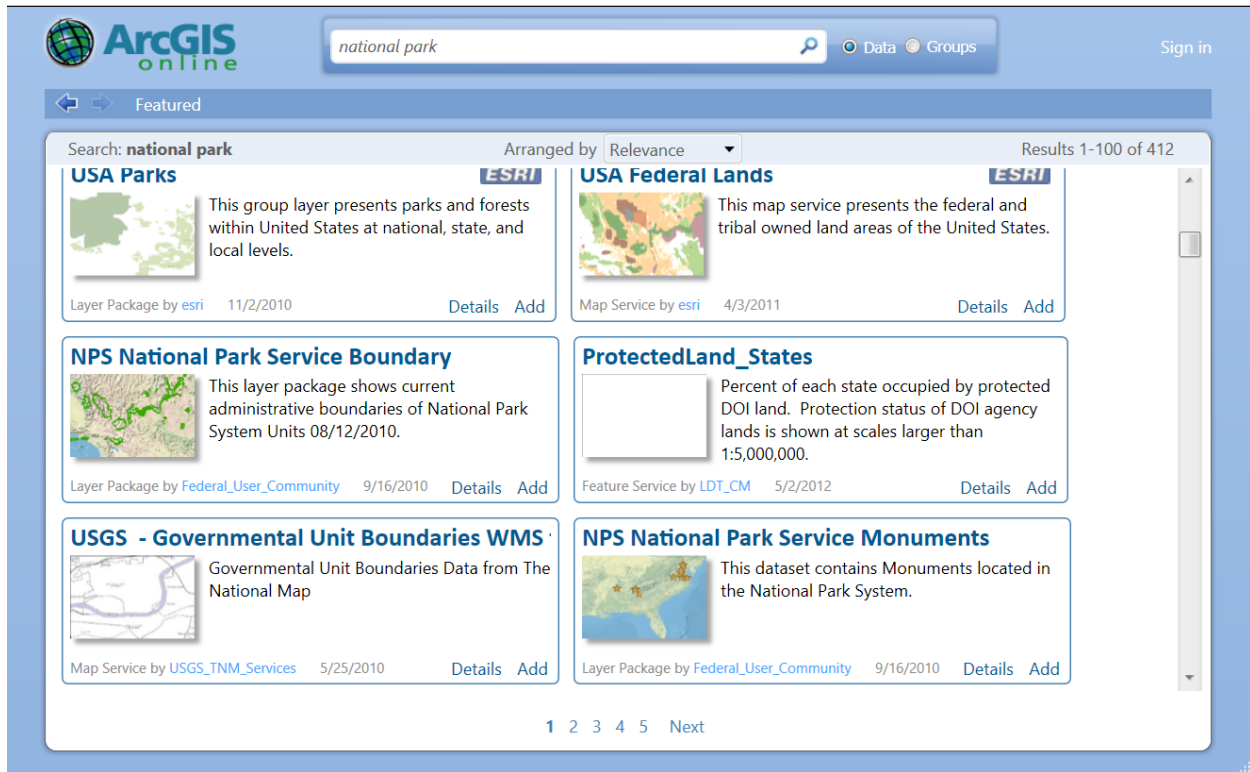
## 17. Wilderness areas

Download data from: <http://www.nationalatlas.gov/mld/wildrnp.html>, and then see Federal Land.

```
> insheet using "E:\Research\NationalAtlas\WildernessArea\data\county_wildarea.txt"
> gen FIPS = statefp10*1000 + countyfp10
> collapse (sum) wildmi2 , by(FIPS)
> rename wildmi2 wildland_mi2
> label variable wildland_mi2 "Wilderness land in square miles"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> sort FIPS
> drop _merge statecode
> replace wildland_mi2 = 0 if missing(wildland_mi2)
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyLand_2010.dta"
> drop _merge
> gen wildland_pct = wildland_mi2/land_mi2*100
> label variable wildland_pct "Wilderness land in percentage (over land area)"
> order FIPS state county , first
> drop land_mi2
> save "E:\Research\NationalAtlas\WildernessArea\WildLand_2010.dta", replace
```

## 18. National parks

Download data from ArcGIS Online. Select 'Add Data' and then 'Add Data from ArcGIS Online...' in ArcMap. Search National Park and then select 'NPS National Park Service Boundary'. Then, see Federal Land.



```
> insheet using "E:\Research\NationalAtlas\NationalPark\data\county_NPSarea.txt", clear
> gen FIPS = statefp10*1000+countyfp10
> collapse (sum) npsmi2 , by(FIPS)
> rename npsmi2 NatParkland_mi2
> label variable NatParkland_mi2 "National Park land in square miles"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> sort FIPS
> drop _merge statecode
> replace NatParkland_mi2 = 0 if missing(NatParkland_mi2)
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyLand_2010.dta"
> drop _merge
> gen NatParkland_pct = NatParkland_mi2/land_mi2*100
> label variable NatParkland_pct "National Park land in percentage (over land area)"
> order FIPS state county , first
> drop land_mi2
> save "E:\Research\NationalAtlas\NationalPark\NationalParkLand_2010.dta", replace
```

## 19. Distance to nearest National Park

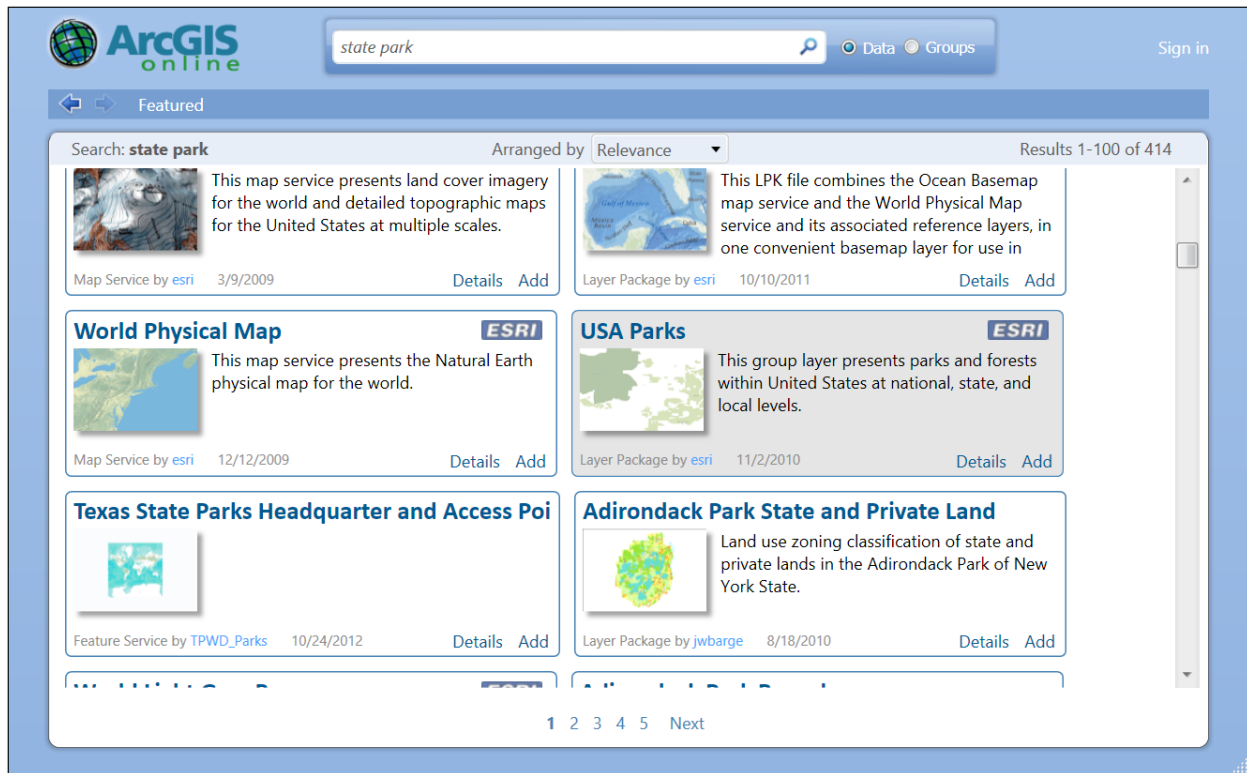
I have used the same shapefile as the one downloaded for 'National Parks' (Section 18). Then, see 'Distance to nearest State Park' (Section 20).

```
> insheet using "E:\Research\NationalAtlas\NationalPark\data\NatPark_distance.csv", clear
> merge 1:1 state county tract blkgrp using "E:\Research\Population\PopBlockGroup\ContiguousUS.dta"
> gen index=population*near_dist
> drop if missing(index)
> drop _merge
> gen FIPS = state*1000+county
> collapse (sum) population index , by(FIPS)
```

```
> gen NatPark_dist = index/population
> drop population index
> replace NatPark_dist = NatPark_dist/1609.34
> label variable NatPark_dist "Distance to nearest national park in miles"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop _merge statecode
> save "E:\Research\NationalAtlas\NationalPark\NationalParkDistance_2010.dta" , replace
```

## 20. Distance to nearest State Park

Download data from ArcGIS Online. Select 'Add Data' and then 'Add Data from ArcGIS Online...' in ArcMap. Search 'State Park' and then select 'USA Parks'.



Then, the data will have 4 groups, categorized by scale. Degroup the data and then just keep the last group, the one with the biggest scale. Export this data to save this shapefile. This shapefile will have USA parks, including federal, state, county and local parks. Since, for now, we are only interested in state parks, we need to open the attribute table and then look for the field that stores this information. In this case, the field 'Featype' has several categories; one of them is 'State park or forest'. Select by attributes and then export the selected data into a different shapefile.

Then we need to project this into a suitable projected coordinate system. Since we are going to calculate distances between county centroids (or blockgroup centroids) and parks, I decided to project the shapefile to the 'USA Contiguous Equidistant Conic' projected coordinate system. Also, we need the distance from counties to state parks, so first I decided to calculate the distance from blockgroup centroids to state parks (polygons, not centroids). Using blockgroup centroids provides a more accurate distance of each group of people to state parks. Also, using the state park polygons instead of the state park centroids provides a more accurate distance since I believe people will start enjoying the amenity (i.e. state park) once they enter or see the park. Therefore, calculating the distance to the polygon, and not the centroid, will be more representative of the real perception.

We need to run the 'Near' tool (ArcToolbox \ Analysis Tools \ Proximity \ Near). Then select the following options:

- Input features: blockgroup centroid shapefile (point)
- Near features: state parks shapefile (polygon)

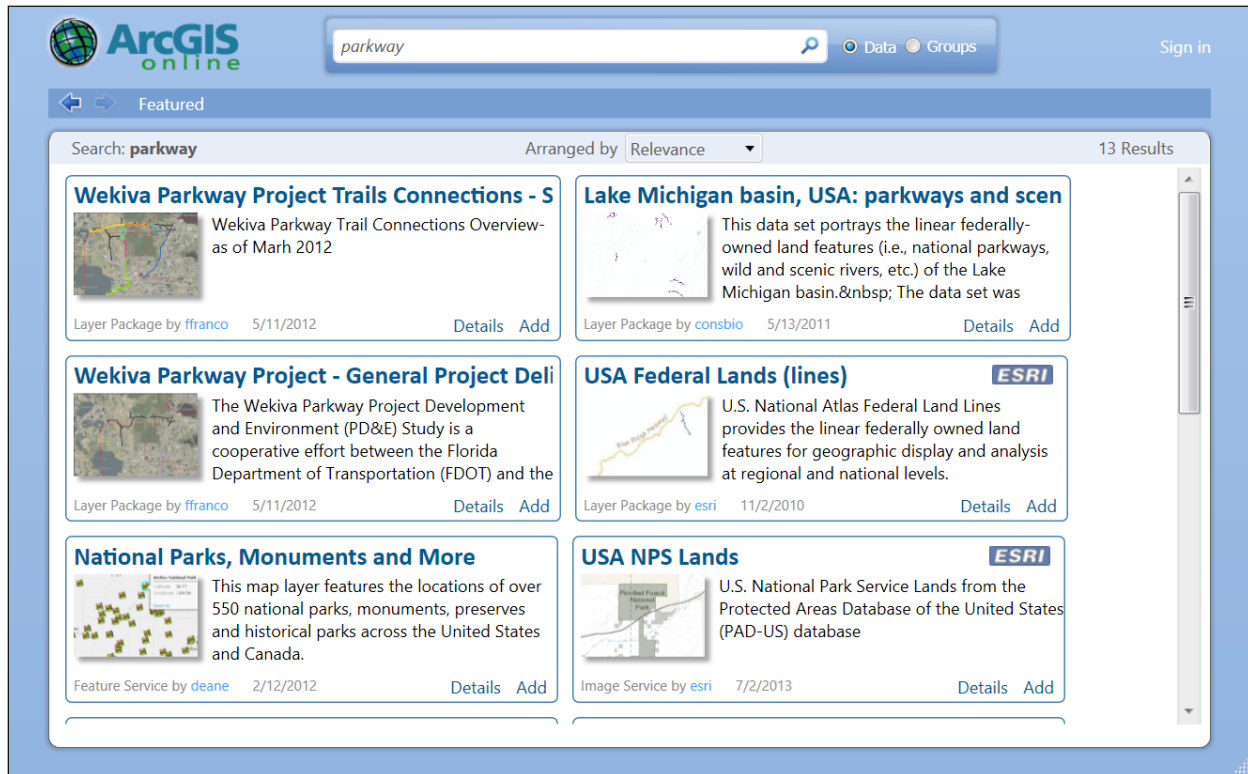
This operation might take several minutes. After doing this we can export the attribute table and then import it into Stata. Then we can run the following commands:

```
> insheet using "E:\Research\NationalAtlas\StatePark\data\centroid_blkgrp_distance.txt", clear
> merge 1:1 state county tract blkgrp using "E:\Research\Population\PopBlockGroup\ContiguousUS.dta"
> browse if _merge==2
> gen index=population*near_dist
> drop if missing(index)
> drop _merge
> gen FIPS = state*1000+county
> collapse (sum) population index , by(FIPS)
> gen statepark_dist = index/population
> drop population index
> replace statepark_dist = statepark_dist/1609.34
> label variable statepark_dist "Distance to nearest state park in miles"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop _merge statecode
> save "E:\Research\NationalAtlas\StatePark\StateParkDistance_2010.dta"
```

## 21. Parkways (Scenic drives) and Scenic Rivers

Parkways are scenic drives. Scenic rivers are not considered scenic drives. According to the Wild and Scenic Rivers Act, selected rivers in free-flowing condition are protected and preserved for the benefit and enjoyment of present and future generations. So, since these scenic rivers may be relatively important, they will be processed in conjunction with scenic drives, and so we will obtain two parameters: one for scenic drives and one for scenic rivers. Also, I will calculate the total mileage of scenic drives and the mileage per square mile of land in the county to standardize this variable. The same will be done for scenic rivers.

Data for parkways and scenic rivers can be downloaded directly from ArcMap. Select 'Add Data' and then 'Add Data from ArcGIS Online...' in ArcMap. Search 'parkway' and then select 'USA Federal Lands (lines)'.



Then, we can intersect the Parkways and Scenic River shapefiles with the county shapefiles in order to split the line shapefiles using the county boundaries. Use the 'Intersect' tool (ArcToolbox \ Analysis Tools \ Overlay \ Intersect) and then select the parkway shapefile or the scenic river shapefile, and the county shapefile. Once we have split both of these line shapefiles using the county boundaries, we can calculate the longitude of each line within each county (Add field, calculate geometry in order to calculate the length in miles and then export the attribute table).

We can import the exported attribute table into Stata and then run the following commands in order to format the data and get a unique dataset that has the following fields:

- Total parkway length in miles
- Total scenic river length in miles
- Parkway length per county land area (mile / mile<sup>2</sup>)
- Scenic river length per county land area (mile / mile<sup>2</sup>)

```
> insheet using "E:\Research\NationalAtlas\Parkway_ScenicRiver\data\NatParkway_length.txt", clear
> gen FIPS = state*1000+county
> keep FIPS length_mil
> order FIPS , first
> rename length_mil NatParkwayLength
> collapse (sum) NatParkwayLength , by(FIPS)
> label variable NatParkwayLength "National Parkway length in miles"
> save "E:\Research\NationalAtlas\Parkway_ScenicRiver\data\NatParkway_length.dta"
```

```
> insheet using "E:\Research\NationalAtlas\Parkway_ScenicRiver\data\NatScenicRiver_length.txt", clear
> gen FIPS = state*1000+county
> keep FIPS length_mil
> order FIPS , first
> rename length_mil ScenicRiverLength
> collapse (sum) ScenicRiverLength , by(FIPS)
> label variable ScenicRiverLength "Scenic rivers length in miles"
> save "E:\Research\NationalAtlas\Parkway_ScenicRiver\data\ScenicRivers_length.dta"
```

```
> use "E:\Research\Census-TIGER\CountyLand_2010.dta", clear
> merge 1:1 FIPS using "E:\Research\NationalAtlas\Parkway_ScenicRiver\data\NatParkway_length.dta"
> drop _merge
> merge 1:1 FIPS using "E:\Research\NationalAtlas\Parkway_ScenicRiver\data\ScenicRivers_length.dta"
> drop _merge
> sort FIPS
> replace NatParkwayLength = 0 if missing(NatParkwayLength)
> replace ScenicRiverLength = 0 if missing(ScenicRiverLength)
> gen NatParkwaybyArea = NatParkwayLength / land_mi2
> gen ScenicRiverbyArea = ScenicRiverLength / land_mi2
> drop land_mi2
> label variable NatParkwaybyArea "Parkway length per county land area, in mile/mile^2"
> label variable ScenicRiverbyArea "Scenic River length per county land area, in mile/mile^2"
> save "E:\Research\NationalAtlas\Parkway_ScenicRiver\Parkway_ScenicRiver2010.dta", replace
```

## 22. Tornado

Data in .csv format can be downloaded from the Storm Prediction Center website

(<http://www.spc.noaa.gov/wcm/>).

Then, we can import each one of these csv datasets into Stata and then run the following commands in order to format the data.

```
> insheet using "E:\Research\NOAA-SPC\Tornadoes\data\50-59_torn.csv", clear
> rename v2 year
> rename v11 f_scale
> drop if v9==2|v9==15|v9==72
> drop if v22==2&v23==0&v24==1
> drop if v22==3&v23==0&v24==1
> rename v9 statecode
> rename v16 s_lat
> rename v17 s_long
> rename v18 e_lat
> rename v19 e_long
> keep year statecode f_scale v25 v26 v27 v28 s_lat s_long e_lat e_long
> save "E:\Research\NOAA-SPC\Tornadoes\data\tornado_1950-1959.dta"
```

Then we can append all of these datasets together.

```
> use "E:\Research\NOAA-SPC\Tornadoes\data\tornado_1950-1959.dta", clear
> append using "E:\Research\NOAA-SPC\Tornadoes\data\tornado_1960-1969.dta"
> append using "E:\Research\NOAA-SPC\Tornadoes\data\tornado_1970-1979.dta"
> append using "E:\Research\NOAA-SPC\Tornadoes\data\tornado_1980-1989.dta"
> append using "E:\Research\NOAA-SPC\Tornadoes\data\tornado_1990-1999.dta"
> append using "E:\Research\NOAA-SPC\Tornadoes\data\tornado_2000-2004.dta"
> append using "E:\Research\NOAA-SPC\Tornadoes\data\tornado_2005-2007.dta"
> append using "E:\Research\NOAA-SPC\Tornadoes\data\tornado_2008.dta"
> append using "E:\Research\NOAA-SPC\Tornadoes\data\tornado_2009.dta"
> save "E:\Research\NOAA-SPC\Tornadoes\data\tornado_data2010.dta"
```

After this, we can merge these files with the FIPS dta file in order to assure that all records have an appropriate FIPS code.

```
> use "E:\Research\NOAA-SPC\Tornadoes\data\tornado_data2010.dta", clear
> gen tornado = 1
> gen tornado_0 = 0
> replace tornado_0 = 1 if f_scale==0
> gen tornado_1 = 0
> replace tornado_1 = 1 if f_scale==1
> gen tornado_2 = 0
> replace tornado_2 = 1 if f_scale==2
> gen tornado_3 = 0
> replace tornado_3 = 1 if f_scale==3
> gen tornado_4 = 0
> replace tornado_4 = 1 if f_scale==4
> gen tornado_5 = 0
```

```

> replace tornado_5 = 1 if f_scale==5
> save "E:\Research\NOAA-SPC\Tornadoes\data\Tornado_data2010.dta", replace
## We need to keep the records that have no county_location information and plot them later into ArcMap. ##
> keep if v25==0&v26==0&v27==0&v28==0
> save "E:\Research\NOAA-SPC\Tornadoes\GIS\Tornado_data2010GIS.dta"
## Then we can keep working with the dataset. ##
> use "E:\Research\NOAA-SPC\Tornadoes\data\Tornado_data2010.dta", clear
> drop if v25==0&v26==0&v27==0&v28==0
> save "E:\Research\NOAA-SPC\Tornadoes\data\Tornado_data2010noGIS.dta", replace
## Then we can work with v25 FIPS codes. ##
> gen FIPS = statecode*1000+v25
> drop if v25==0
> merge m:1 FIPS using "E:\Research\FIPS\FIPS.dta"
## Note: Now look for records that did not match and replace their FIPS codes when they are no longer in use.
For instance, FIPS 12025 (Dade County) is no longer used, it was replaced by 12086 (Miami-Dade County). After
this has been done we can continue formatting the dataset. ##
> browse if _merge==1
> replace FIPS = 12086 if FIPS==12025
> replace FIPS = 46041 if FIPS==46001
> replace FIPS = 46071 if FIPS==46131
> replace FIPS = 51800 if FIPS==51123
> replace FIPS = 51083 if FIPS==51780
> replace FIPS = 51036 if FIPS==51037&year<1979
> replace FIPS = 51037 if FIPS==51039
> drop county state _merge
> drop if missing(tornado)
> merge m:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==1|_merge==2
> collapse (sum) tornado tornado_0 tornado_1 tornado_2 tornado_3 tornado_4 tornado_5 , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==2
> drop statecode _merge
> save "E:\Research\NOAA-SPC\Tornadoes\by_v\Tornado2010_v25.dta"
## Then we can work with v26 FIPS codes. ##
> use "E:\Research\NOAA-SPC\Tornadoes\data\Tornado_data2010noGIS.dta", clear
> gen FIPS = statecode*1000+v26
> drop if v26==0
> merge m:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> browse if _merge==1
> replace FIPS = 12086 if FIPS==12025
> drop county state _merge
> drop if missing(tornado)
> merge m:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==1|_merge==2
> collapse (sum) tornado tornado_0 tornado_1 tornado_2 tornado_3 tornado_4 tornado_5 , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop _merge==2
> drop statecode _merge
> save "E:\Research\NOAA-SPC\Tornadoes\by_v\Tornado2010_v26.dta"
## Working with v27 FIPS codes. ##
> use "E:\Research\NOAA-SPC\Tornadoes\data\Tornado_data2010noGIS.dta", clear
> gen FIPS = statecode*1000+v27
> drop if v27==0
> merge m:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> browse if _merge==1
> drop county state _merge
> drop if missing(tornado)
> merge m:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==1|_merge==2
> collapse (sum) tornado tornado_0 tornado_1 tornado_2 tornado_3 tornado_4 tornado_5 , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==2
> drop statecode _merge
> save "E:\Research\NOAA-SPC\Tornadoes\by_v\Tornado2010_v27.dta"
## Working with v28 FIPS codes. ##
> use "E:\Research\NOAA-SPC\Tornadoes\data\Tornado_data2010noGIS.dta", clear
> gen FIPS = statecode*1000+v28
> drop if v28==0
> merge m:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> browse if _merge==1
> drop county state _merge
> drop if missing(tornado)
> merge m:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==1|_merge==2

```



```
> collapse (sum) tornado tornado_0 tornado_1 tornado_2 tornado_3 tornado_4 tornado_5 , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==2
> drop statecode _merge
> save "E:\Research\NOAA-SPC\Tornadoes\by_v\Tornado2010_v28.dta"
```

Now that we have created datasets for v25, v26, v27 and v28, we can append them together and then collapse the data into a single dta file. Remember that we will need to add some tornado information from the observations that will be plotted on ArcMap.

```
> use "E:\Research\NOAA-SPC\Tornadoes\by_v\Tornado2010_v25.dta", clear
> append using "E:\Research\NOAA-SPC\Tornadoes\by_v\Tornado2010_v26.dta"
> append using "E:\Research\NOAA-SPC\Tornadoes\by_v\Tornado2010_v27.dta"
> append using "E:\Research\NOAA-SPC\Tornadoes\by_v\Tornado2010_v28.dta"
> collapse (sum) tornado tornado_0 tornado_1 tornado_2 tornado_3 tornado_4 tornado_5 , by(FIPS county state)
> order state , after(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> sort FIPS
> replace tornado = 0 if missing(tornado)
> replace tornado_0 = 0 if missing(tornado_0)
> replace tornado_1 = 0 if missing(tornado_1)
> replace tornado_2 = 0 if missing(tornado_2)
> replace tornado_3 = 0 if missing(tornado_3)
> replace tornado_4 = 0 if missing(tornado_4)
> replace tornado_5 = 0 if missing(tornado_5)
> save "E:\Research\NOAA-SPC\Tornadoes\Tornado2010.dta"
```

Now we need to plot the observations that did not have county location information. We can do this by opening the dta files and exporting them as Excel files. After adding the Excel file to ArcMap, we can convert the starting point and ending point to a line which defines the tornado path for each one of the tornadoes by using the tool ‘Points to Line’ (ArcToolbox \ Data Management Tools \ Features \ Points to Line). After doing this we can split the lines using the county boundary information by using the tool ‘Intersect’ (ArcToolbox \ Analysis Tools \ Overlay \ Intersect).

Once we have identified the additional counties that intersected the tornadoes’ path we can add this information to the ‘Tornado2010\_collapse\_sum.dta’ file by running the command ‘edit’ and increasing the number of tornadoes by 1.

Then, creating the final dta file is just a matter of dividing by the number of years and labeling the variables.

```
> use "E:\Research\NOAA-SPC\Tornadoes\Tornado2010.dta", clear
> replace tornado = tornado/(2009-1949)
> replace tornado_0 = tornado_0/(2009-1949)
> replace tornado_1 = tornado_1/(2009-1949)
> replace tornado_2 = tornado_2/(2009-1949)
> replace tornado_3 = tornado_3/(2009-1949)
> replace tornado_4 = tornado_4/(2009-1949)
> replace tornado_5 = tornado_5/(2009-1949)
> label variable tornado "Average number of tornadoes per annum, 1950 - 2009"
> label variable tornado_0 "Average number of 0 f-scale tornadoes per annum, 1950 - 2009"
> label variable tornado_1 "Average number of 1 f-scale tornadoes per annum, 1950 - 2009"
> label variable tornado_2 "Average number of 2 f-scale tornadoes per annum, 1950 - 2009"
> label variable tornado_3 "Average number of 3 f-scale tornadoes per annum, 1950 - 2009"
> label variable tornado_4 "Average number of 4 f-scale tornadoes per annum, 1950 - 2009"
> label variable tornado_5 "Average number of 5 f-scale tornadoes per annum, 1950 - 2009"
> save "E:\Research\NOAA-SPC\Tornadoes\Tornado2010.dta" , replace
```

## 23. Property damage from hazard events

Data can be downloaded from the “Spatial Hazard Events and Losses Database for the United States”, from the University of South Carolina (<http://webra.cas.sc.edu/hvri/products/sheldus.aspx>). Once you click on “Data



download" you will be prompted to a website where you need to select the search query. Select "Search by Hazard type, location and date", "All State", "All counties", and "All hazard types". Select the date range per year, since downloads are restricted to 200,000 records. Once you get the results, remember to include the inflation adjusted losses. We will cover data from the decade previous to the analysis, i.e. 2000 to 2009, so we will need to adjust inflation to 2009.

The SHELDUS website provides very useful information; however it may not be very accurate since it appears that it divides the property damage equitably among counties, which may lead to overestimating and underestimating damages.

Once we have downloaded the csv files for years 2000 to 2009, we can import them into Stata, append them and collapse them. Run the following commands:

```
> insheet using "E:\Research\SHELDUS\PropertyDamage\data\results2000.csv", clear
> gen year = 2000
> rename fips_code FIPS
> keep FIPS year property_damage_adjusted_2009
> save "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2000.dta"
> insheet using "E:\Research\SHELDUS\PropertyDamage\data\results2001.csv", clear
> gen year = 2001
> rename fips_code FIPS
> keep FIPS year property_damage_adjusted_2009
> save "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2001.dta"
> insheet using "E:\Research\SHELDUS\PropertyDamage\data\results2002.csv", clear
> gen year = 2002
> rename fips_code FIPS
> keep FIPS year property_damage_adjusted_2009
> save "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2002.dta"
> insheet using "E:\Research\SHELDUS\PropertyDamage\data\results2003.csv", clear
> gen year = 2003
> rename fips_code FIPS
> keep FIPS year property_damage_adjusted_2009
> save "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2003.dta"
> insheet using "E:\Research\SHELDUS\PropertyDamage\data\results2004.csv", clear
> gen year = 2004
> rename fips_code FIPS
> keep FIPS year property_damage_adjusted_2009
> save "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2004.dta"
> insheet using "E:\Research\SHELDUS\PropertyDamage\data\results2005.csv", clear
> gen year = 2005
> rename fips_code FIPS
> keep FIPS year property_damage_adjusted_2009
> save "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2005.dta"
> insheet using "E:\Research\SHELDUS\PropertyDamage\data\results2006.csv", clear
> gen year = 2006
> rename fips_code FIPS
> keep FIPS year property_damage_adjusted_2009
> save "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2006.dta"
> insheet using "E:\Research\SHELDUS\PropertyDamage\data\results2007.csv", clear
> gen year = 2007
> rename fips_code FIPS
> keep FIPS year property_damage_adjusted_2009
> save "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2007.dta"
> insheet using "E:\Research\SHELDUS\PropertyDamage\data\results2008.csv", clear
> gen year = 2008
> rename fips_code FIPS
> keep FIPS year property_damage_adjusted_2009
> save "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2008.dta"
> insheet using "E:\Research\SHELDUS\PropertyDamage\data\results2009.csv", clear
> gen year = 2009
> rename fips_code FIPS
> keep FIPS year property_damage_adjusted_2009
> save "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2009.dta"
> use "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2000.dta", clear
> append using "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2001.dta"
> append using "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2002.dta"
> append using "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2003.dta"
> append using "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2004.dta"
```

```
> append using "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2005.dta"
> append using "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2006.dta"
> append using "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2007.dta"
> append using "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2008.dta"
> append using "E:\Research\SHELDUS\PropertyDamage\data\propertydmg2009.dta"
> collapse (sum) property_damage_adjusted_2009 , by(FIPS)
> label variable FIPS ""
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==1
> replace property_damage_adjusted_2009 = 0 if missing(property_damage_adjusted_2009)
> drop _merge statecode
> sort FIPS
> replace property_damage_adjusted_2009 = property_damage_adjusted_2009/1000
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyLand_2010.dta"
> drop _merge
> gen property_damage_per_area = property_damage_adjusted_2009/land_mi2
> drop land_mi2
> label variable property_damage_per_area "Property damage from hazard events (2000-2009) in $000s/mile^2 of
land, adjusted 2009"
> label variable property_damage_adjusted_2009 "Property damage from hazard events (2000-2009) in $000s,
adjusted 2009"
> save "E:\Research\SHELDUS\PropertyDamage\PropertyDamage2010.dta" , replace
```

## 24. Seismic hazard

The latest data was downloaded from the following website:

<http://www.nationalatlas.gov/maplayers.html?openChapters=chpgeol#chpgeol>

<http://www.nationalatlas.gov/mld/seihazp.html>

<http://www.nationalatlas.gov/atlasftp.html#seihazp>

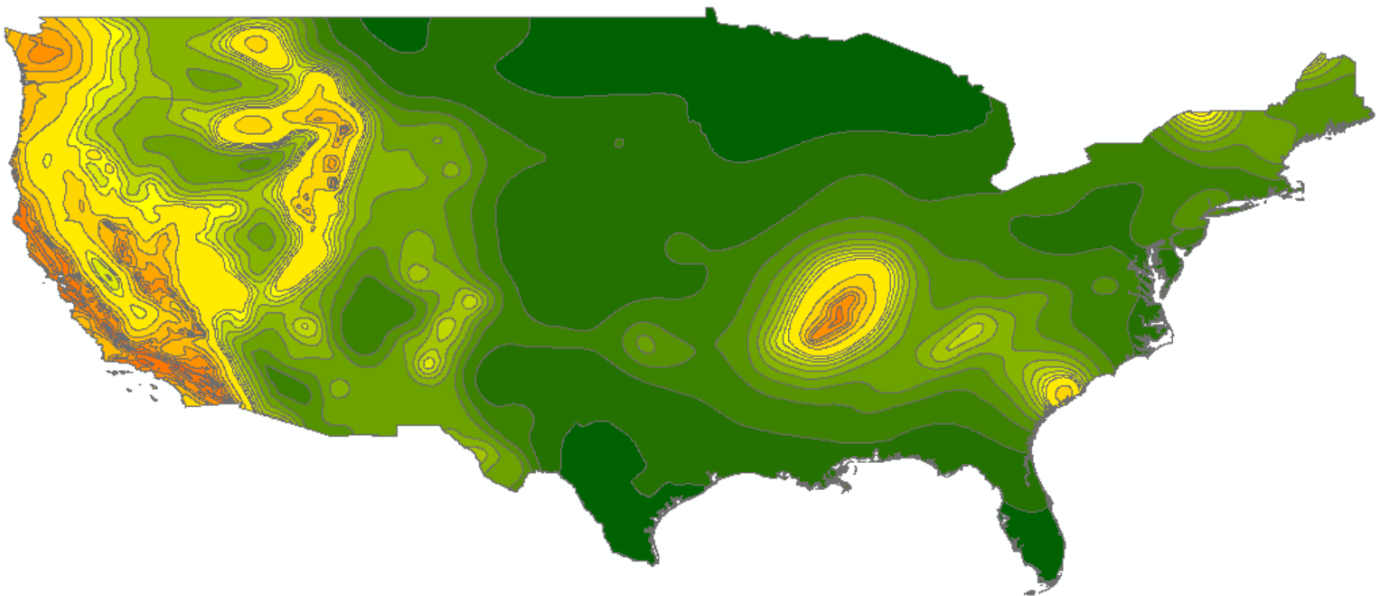
Once this data is downloaded, you will notice that the field Percent\_g stands for the Seismic Hazard Index. The seismic hazard index is expressed as the peak horizontal ground acceleration experienced in that area as percent of gravity (with a 10% probability of exceedance in 50 years).

|   | FID | Shape * | VALLEY | Percent_g |
|---|-----|---------|--------|-----------|
| ▶ | 0   | Polygon | 0      | 15 - 20   |
|   | 1   | Polygon | 0      | 15 - 20   |
|   | 2   | Polygon | 0      | 4 - 5     |
|   | 3   | Polygon | 0      | 4 - 5     |
|   | 4   | Polygon | 0      | 4 - 5     |
|   | 5   | Polygon | 0      | 4 - 5     |
|   | 6   | Polygon | 0      | 4 - 5     |
|   | 7   | Polygon | 0      | 4 - 5     |
|   | 8   | Polygon | 0      | 4 - 5     |
|   | 9   | Polygon | 0      | 4 - 5     |
|   | 10  | Polygon | 0      | 5 - 6     |
|   | 11  | Polygon | 0      | 5 - 6     |
|   | 12  | Polygon | 0      | 8 - 9     |
|   | 13  | Polygon | 0      | 9 - 10    |
|   | 14  | Polygon | 0      | 80 - 100  |
|   | 15  | Polygon | 0      | 80 - 100  |
|   | 16  | Polygon | 0      | 80 - 100  |
|   | 17  | Polygon | 0      | 80 - 100  |

Since the field 'Percent\_g' is stored as text, we need to create a new field in order to store the seismic hazard index as numeric. Since we are dealing with intervals, I assumed that the mean index will be the interval midpoint. In order to do this, I did the following:

- (1) Selection \ Select By Attributes...
- (2) Select the following options from the pop-up window:
  - a. Layer: 'Name of the layer' (In this case, the name of the layer was 'Indice')
  - b. Method: Create a new selection
  - c. Percent\_g
  - d. SELECT \* FROM 'Indice' WHERE:
  - e. [Percent\_g] = '> 100'
  - f. Click 'OK'
- (3) Once a single category of 'Percent\_g' is selected, you can export the selection as a different dataset. Right click on the layer, and then click 'Data \ Export Data...'
- (4) Select the following options from the pop-up window:
  - a. Export: 'Selected features'
  - b. Use the same coordinate system as: this layer's source data.
  - c. Output feature class: 'path\filename'
- (5) Add a field in the newly created shapefile.
- (6) Right click the new field and click on 'Field Calculator...'
- (7) Type in the white space in the pop-up window the number we wish to assign to this field. This number will be equal to the interval midpoint of 'Percent\_g'.
- (8) Repeat this for every category within 'Percent\_g'.
- (9) Once we have a shapefile for each one of the 'Percent\_g' categories, we can merge them all in a single shapefile. Select from ArcToolbox the following option: Data Management Tools \ General \ Merge


Now that we have a single file with a numeric field that stores the information of seismic hazard index, we can take this information to the county level.



Since the information is spatially distributed, we need to find a way to integrate this information either by area or by population. It would not be suitable to integrate it by area, since the population is not evenly distributed and it may be the case that households chose to concentrate in such a pattern in which the average person would experience a higher or lower risk than the area-weighted mean risk across the county. In this sense, I used information about population by blockgroup and used the blockgroup boundary file in order to first, estimate the risk the blockgroup centroid would experience, and then estimate the population-weighted average at the county level.

The blockgroup population-weighted centroid file can be downloaded from the following census website:  
<http://www.census.gov/geo/reference/centersofpop.html>

After downloading this txt file, we can import it into ArcGIS.

- (1) Add the txt file by clicking on the icon with the shape of a plus sign. 
- (2) Once this txt file is imported, right click it and then click on 'Display XY Data...'
  - a. X Field: LONGITUDE
  - b. Y Field: LATITUDE
  - c. Coordinate System of Input Coordinates:  
Geographic Coordinate System GCS\_North\_American\_1983
- (3) By doing this, we will have plotted the table as a layer.
- (4) Then we need to save it as a shapefile. Right click on the newly displayed layer, click on 'Data' and then 'Export Data...'
- (5) Add this recently save shapefile.

Having these two shapefiles, we can spatially join them (the blockgroup population-weighted centroid and the seismic hazard index). We need to assign a given seismic hazard index to each centroid, so I did the following:

- (1) Right click the blockgroup population-weighted centroid shapefile.
- (2) Select Join and Relates \ Join...
- (3) Then select: 'Join data from another layer based on spatial location' and select the seismic hazard index shapefile and also select the 'is closest to it' option. I chose the 'is closest to it' option because it may be the case that some points might fall outside of the seismic hazard index extension and thus they won't be assigned any index. This may take several minutes.

Once we've obtained the new blockgroup centroid shapefile with seismic hazard index information, we can export the attribute table as a text file.

- (1) Right click the new blockgroup centroid shapefile.
- (2) Select Open Attribute Table...
- (3) Click on the far top left icon on the Attribute Table; click 'Export...'
  - a. Export: All records
  - b. Output table: 'path\filename'



Now that we have the .txt file, we can import it into Stata. Run File \ Import \ Text data created by a spreadsheet. Then run the following commands in order to keep the necessary variables.

```
> insheet using "E:\Research\NationalAtlas\Seismic_hazard\seismic_centroid.txt", clear
> keep statefp countyfp population seismic_ha
> gen FIPS = statefp*1000+countyfp
> order FIPS , first
```

```
> drop statefp countyfp
> sort FIPS
> gen popseismic = population*seismic_ha
```

Then, we can collapse the information in order to obtain the seismic hazard index at the county level.

```
> collapse (sum) popseismic population, by(FIPS)
> gen seismic_hazard = popseismic/population
> keep FIPS seismic_hazard
> label variable seismic_hazard "Seismic hazard index"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> save "E:\Research\NationalAtlas\Seismic_hazard\seismic_hazard.dta"
```

## 25. Number of Earthquakes

Earthquake data can be downloaded directly from the following website:

<http://earthquake.usgs.gov/earthquakes/map/>.

This website is very useful; one can select different parameters, such as magnitude, location, and period of analysis. Since the number of earthquakes can be massive, I recommend processing one year at a time. Since this website can provide a very broad range of information, it is important to narrow it down. For instance, I am only downloading information regarding earthquakes with a magnitude or 4 or higher. This is because, according to the USGS website (<http://www.usgs.gov/faq/?q=categories/9829/3318>), "damage does not usually occur until the earthquake magnitude reaches somewhere above 4 or 5."

You can download data by year (downloading by decade will be too much to handle for the website) and select only earthquakes with magnitude between 4 and 10. Do not select any spatial range since you may lose any earthquakes just outside the boundary of analysis if you decided to do so.

It seems that the data from the earthquake USGS website provides more information than the National Atlas. For instance, the shapefile from the National Atlas (<http://nationalatlas.gov/atlasftp.html#quksigx>) has 5,402 records while the dataset downloaded from the USGS website for year 2009 has 8,862 records.

One of the challenges with working with earthquakes data is that there are spatial externalities and assigning an earthquake value to a single county might lead to not appropriately representing the information. However, since this might have been previously considered in other studies, I am deciding to count the number of earthquakes, as well as a number of indices which will depend on the distance to the epicenter and the magnitude of the earthquake. One of the reasons for using the indices is that, for instance, if a given coastal county experiences several high-magnitude earthquakes, but all of their epicenters are located below the sea, then counting the number of earthquakes will yield a number that fully represents the earthquake perception people in that county might feel. In these cases, having an index would be useful.

Given that earthquakes can be very different in magnitude, it is also vital to discriminate them by their magnitude. For instance, if we merely count the number of earthquakes it could be the case that a county gets a value of 10 and another one gets a value of 1, even though the former experienced 10 earthquakes with a magnitude equal to 4 (which is felt but shouldn't damage structures), whereas the latter experienced 1 earthquake with a magnitude equal to 9 (which is incredibly powerful).

Since there is a spatial component, I decided to use the inverse distance as an input for the indices. In this sense, if an earthquake happened very far away, we would obtain a negligible value and if it happened close enough, we would obtain values closer to 1. The steps followed to create the indices are the following:

- (1) Classify the earthquakes by magnitude, creating 4 classes:
  - a.  $4 - < 5$
  - b.  $5 - < 6$
  - c.  $6 - < 7$
  - d.  $7 - < 8$
- (2) Calculate the inverse distance from each county population-weighted centroid to each epicenter, by class.
- (3) Sum the inverse distances for each county, by class.
- (4) Since people might 'forget' about earthquakes, we will calculate the index first for the earthquakes that happened within the last decade (2000-2009), then within the last two decades (1990-2009), within the last 4 decades (1970-2009), and finally using the whole spectrum of available data.


Once I had imported the whole dataset from 1970 to 2009, I ran the following commands:

```
> gen year = substr(time,1,4)
> destring year , replace
> order year , first
> keep if latitude<=52&latitude>=23
> keep if longitude<=-64&longitude>=-128
```

Then, in order to analyze the earthquakes from 2000 to 2009 I split the dataset. I needed to work with the dataset on Stata, then import the table to ArcGIS, and then once I had projected the shapefile, I needed to calculate the distances on 'Geospatial Modelling Environment | SpatialEcology.com'. This is the example for working with years 2000-2009 and magnitude 6-7. I ran the following commands in Stata:

```
> use "E:\Research\USGS\Earthquakes\earthquakes1970-2009.dta", clear
> keep if year>1999
> keep if mag>=6&mag<7
> save "E:\Research\USGS\Earthquakes\earthquakes2000-2009_6-7.dta"
> export excel using "E:\Research\USGS\Earthquakes\Excel\earthquakes2000-2009_6-7.xls", firstrow(variables)
```

Now that we have an Excel file for years 2000-2009 and magnitude 6-7, we can import it into ArcGIS. Follow these steps to do so.

- (1) Add the exported Excel file by clicking on the icon with the shape of a plus sign. 
- (2) Once this table is imported, right click it and then click on 'Display XY Data...'
  - a. X Field: Longitude
  - b. Y Field: Latitude
  - c. Coordinate System of Input Coordinates:  
Geographic Coordinate System WGS84. I am assuming that this is the GCS since this data was downloaded from a worldwide dataset.
- (3) By doing this, we will have plotted the table as a layer.
- (4) Then we need to save it as a shapefile. However, since we will need to calculate distances it is best to project it simultaneously. Run the tool Project, which can be found on ArcToolbox \ Data Management Tools \ Projections and Transformations \ Feature \ Project. Click on this tool and then choose the following settings:
  - a. Input dataset or Feature class: newly plotted layer
  - b. Output dataset or Feature class: E:\Research\GIS\Shapefiles\Earthquakes\quake2000to2009\_6to7
  - c. Output coordinate system: USA Contiguous Equidistant Conic
- (5) Then click ok and we will have obtained a layer

Choosing the right projected coordinate system is very important. Since the data is initially plotted in geographic coordinate systems, we need to project it on a plane in order to calculate distances. I am using the USA Contiguous Equidistant Conic projection since it should provide distort the least the distances, given that the area of analysis is the contiguous U.S.

After doing so, we need to calculate the distances between epicenters and county centroids. In order to do so, we need to download the software 'Geospatial Modelling Environment' (<http://www.spatial ecology.com/gme/>). GME uses the open source statistical software 'R'. Therefore, it is extremely recommended that you install 'R' before working with GME. Once you open R (or R-Studio), run the following command to install the necessary packages:

```
> install.packages(pkgs=c("CircStats", "odesolve", "coda", "deldir", "igraph", "RandomFields", "ks"))
```

Then, after downloading and installing GME, test the connection to R by going to Help, then Citation. You should see citation info for R and GME.

After having successfully installed GME, open it and follow the next steps in order to calculate the distances between epicenters and county centroids.

- (1) Select 'pointdistances' from the Commands list.
- (2) Select the following options:
  - a. In: county centroide shapefile, e.g.  
E:\Research\GIS\Shapefiles\Population\centroid\_county\_2010\_proj
  - b. fld: FIPS
  - c. out: distance matrix .txt file, e.g.  
E:\Research\USGS\Earthquakes\distances\distance2000to2009\_6to7.txt
  - d. Optional: in2: earthquake shapefile, e.g.  
E:\Research\GIS\Shapefiles\Earthquakes\quake2000to2009\_6to7
  - e. Optional: fld2: OBJECTID
  - f. Optional: format: 2D

Then run the command and once the process has been completed, you'll find a .txt file in the output path. Now we can import this .txt file into Stata and calculate the sum of inverse distances index<sup>4</sup> by running the following commands:

```
> insheet using "E:\Research\USGS\Earthquakes\distance\distance2000to2009_6to7.txt"
> rename uid FIPS
> replace v2 = v2/1609.34
> replace v3 = v3/1609.34
> replace v4 = v4/1609.34
> replace v5 = v5/1609.34
> replace v6 = v6/1609.34
> replace v7 = v7/1609.34
> replace v8 = v8/1609.34
> replace v9 = v9/1609.34
> replace v10 = v10/1609.34
> replace v11 = v11/1609.34
> replace v12 = v12/1609.34
> replace v13 = v13/1609.34
> replace v14 = v14/1609.34
> replace v15 = v15/1609.34
> replace v16 = v16/1609.34
> replace v17 = v17/1609.34
> replace v2 = 1/(v2*v2)*10^8
```

---

<sup>4</sup> The distance is calculated in meters, so since the index will be expressed in miles<sup>-1</sup>, we need to convert units as well.



```
> replace v3 = 1/(v3*v3)*10^8
> replace v4 = 1/(v4*v4)*10^8
> replace v5 = 1/(v5*v5)*10^8
> replace v6 = 1/(v6*v6)*10^8
> replace v7 = 1/(v7*v7)*10^8
> replace v8 = 1/(v8*v8)*10^8
> replace v9 = 1/(v9*v9)*10^8
> replace v10 = 1/(v10*v10)*10^8
> replace v11 = 1/(v11*v11)*10^8
> replace v12 = 1/(v12*v12)*10^8
> replace v13 = 1/(v13*v13)*10^8
> replace v14 = 1/(v14*v14)*10^8
> replace v15 = 1/(v15*v15)*10^8
> replace v16 = 1/(v16*v16)*10^8
> replace v17 = 1/(v17*v17)*10^8
> egen index_00_09_6_7 = rowtotal(_all)
> replace index_00_09_6_7=index_00_09_6_7-FIPS
> replace index_00_09_6_7=index_00_09_6_7/10^8
> keep FIPS index_00_09_6_7
> label variable index_00_09_6_7 "Earthquake index in miles^-2, 6 to 7 magnitude, 2000 to 2009"
```

Stata might not be able to read a dataset because it is too large. When this happens we can rely on R. For instance, I was not able to read the distance matrix for earthquakes from 1970 to 2009 of magnitude between 4 and 5. However, I was able to import this .txt file into R. I ran the following commands:

```
> setwd("E:/Research/USGS/Earthquakes/distance")
> data <- read.table("distance1970to2009_4to5.txt",header=T,sep=",")
> justdata <- data[c(-1)]
> data.miles <- justdata/1609.34
> data.inv.sq <- 1/data.miles^2
> index <- rowSums(data.inv.sq,na.rm=F,dims=1)
> FIPS <- data[c(1)]
> data.final <- cbind(FIPS,index)
> write.csv(data.final,file="index.csv")
```

Then, we can import this .csv file into Stata and work with it as usual.

After creating all the index files for each magnitude class, we can merge them together. Then, we also need to have the ‘benchmark’ earthquake variable, i.e. ‘number of earthquakes by county’. I also calculated this variable by magnitude and years.

I plotted the earthquakes from a given magnitude class (e.g. 5 to 6) and a given period (e.g. 2000 to 2009) and the county boundaries. Then by spatially joining these two shapefiles we can obtain the number of earthquakes that occurred in each county and then export the table of attributes of the joined shapefile. After exporting this table, we need to import it into Stata and format it by running the following commands:

```
> gen FIPS = state*1000 + county
> rename count_ quake_90_09_5_6
> keep FIPS quake_90_09_5_6
> sort FIPS
> order FIPS , first
> label variable quake_90_09_5_6 "Earthquakes 5 to 6 magnitude, 1990 to 2009"
```

Then we can save this dta file, and merge it with the other dta files that stored the number of earthquakes and finally merge it with the earthquake index dta file.

This website may be of future use: <http://www.isc.ac.uk/registries/>



## 26. Land Cover Diversity

The latest data is collected in December 2002. Data can be accessed from

<http://nationalatlas.gov/atlasftp.html?openChapters=chpgeol%2Cchpbio#chpbio>. Raw data is stored in C:\Data\Land Cover Diversity. Raw data format is .TFW, and I have no idea how to deal with it in GIS or STATA.

**2002 data is too old and in .TFW format.**

## 27. NPDES effluent dischargers

This variable describes the number of effluent discharge facilities per county by counting the number of permits issued during the previous 10 years (e.g. for year 2010 I would count the permits issued from 2001 to 2010). Effluent discharger data is collected from EPA-TRI <http://www.epa.gov/enviro/facts/pcs-icis/index.html>. You can customize your selection in <http://www.epa.gov/enviro/facts/pcs-icis/customized.html>.

In order to customize the data, I selected the 'Facility Information' table in Step 1. Then proceed to Step 2; in this step you can select which tables you want from this dataset, check the v\_pcs\_permit\_facilities (facility that discharges pollutants into U.S. waters). Then proceed to Step 3 and check 'NPDES', 'City Name', 'County Name', 'County Code', 'Latitude', 'Longitude', 'Location City', 'Location State', 'Permit Expired Date', 'Permit Issued Date', 'Datum', and 'Code Expansion for Datum'. Then proceed to Step 4 and click on Output to CSV file.

Now that the csv file has been downloaded, we can import this file into Stata and then run the following commands to format the variables in more suitable names:

```
> rename v_pcs_permit_facilitiesnpdes NPDES
> rename v_pcs_permit_facilitiescity_name City
> rename v_pcs_permit_facilitiescounty_na County
> rename v_pcs_permit_facilitiescounty_co Countycode
> rename v_pcs_permit_facilitieslatitude Latitude
> rename v_pcs_permit_facilitieslongitude Longitude
> rename v_pcs_permit_facilitiesloc_city LocCity
> rename v_pcs_permit_facilitiesloc_state State
> rename v_pcs_permit_facilitiespermit_ex ExpirationDate
> rename v_pcs_permit_facilitiespermit_is IssuanceDate
> rename v_pcs_permit_facilitiesdatum DatumCode
> rename codeexpansionfordatum Datum
> label variable NPDES "NPDES"
> label variable City "City name"
> label variable County "County name"
> label variable Countycode "County code"
> label variable Latitude "Latitude"
> label variable Longitude "Longitude"
> label variable LocCity "Location City"
> label variable State "State"
> label variable ExpirationDate "Permit Expiration Date"
> label variable IssuanceDate "Permit Issuance Date"
> label variable DatumCode "Datum Code"
> label variable Datum "Datum"
```

However, it may be the case that some of the fields have the character '“' only once. When this happens in a csv file Stata takes all the following characters as if they were a single record until another '”' is found. This creates several errors in the dataset, since it could be the case that it would store several observations in a single record. In order to avoid this error from happening one could open the csv file in Wordpad and then replace all the '“' for '“'.

Then, in order to analyze if the facility had a valid permit during the previous decade one needs to create a variable that stores the expiration date and issuance date as 'date' and not as 'string'. Thus, we need to run the following commands:

```
> gen ExpDate = date(ExpirationDate,"YMD")
> format ExpDate %d
> gen IssDate = date(IssuanceDate,"YMD")
> format IssDate %d
> drop ExpirationDate IssuanceDate
> rename ExpDate ExpirationDate
> rename IssDate IssuanceDate
> label variable ExpirationDate "Permit Expiration Date"
> label variable IssuanceDate "Permit Issuance Date"
```

Then we can drop the observations that have an ExpirationDate prior to an IssuanceDate since this is obviously inconsistent.

```
> drop if ExpirationDate<IssuanceDate
```

There are some observations that have no data in the State variable; however, the first two letters in the NPDES variable stand for the State abbreviation, so we can extract these first two letters and save them as a new variable. You can run the following command:

```
> gen Stateabb = substr(NPDES,1,2)
> drop State
> rename Stateabb State
> label variable State "State"
> gen Statecode = 0
> label variable Statecode "State code"
> replace Statecode = 1 if State=="AL"
> replace Statecode = 4 if State=="AZ"
> replace Statecode = 5 if State=="AR"
> replace Statecode = 6 if State=="CA"
> replace Statecode = 8 if State=="CO"
> replace Statecode = 9 if State=="CT"
> replace Statecode = 10 if State=="DE"
> replace Statecode = 11 if State=="DC"
> replace Statecode = 12 if State=="FL"
> replace Statecode = 13 if State=="GA"
> replace Statecode = 16 if State=="ID"
> replace Statecode = 17 if State=="IL"
> replace Statecode = 18 if State=="IN"
> replace Statecode = 19 if State=="IA"
> replace Statecode = 20 if State=="KS"
> replace Statecode = 21 if State=="KY"
> replace Statecode = 22 if State=="LA"
> replace Statecode = 23 if State=="ME"
> replace Statecode = 24 if State=="MD"
> replace Statecode = 25 if State=="MA"
> replace Statecode = 26 if State=="MI"
> replace Statecode = 27 if State=="MN"
> replace Statecode = 28 if State=="MS"
> replace Statecode = 29 if State=="MO"
> replace Statecode = 30 if State=="MT"
> replace Statecode = 31 if State=="NE"
> replace Statecode = 32 if State=="NV"
> replace Statecode = 33 if State=="NH"
> replace Statecode = 34 if State=="NJ"
> replace Statecode = 35 if State=="NM"
> replace Statecode = 36 if State=="NY"
> replace Statecode = 37 if State=="NC"
> replace Statecode = 38 if State=="ND"
> replace Statecode = 39 if State=="OH"
> replace Statecode = 40 if State=="OK"
> replace Statecode = 41 if State=="OR"
> replace Statecode = 42 if State=="PA"
> replace Statecode = 44 if State=="RI"
> replace Statecode = 45 if State=="SC"
```

```
> replace Statecode = 46 if State=="SD"
> replace Statecode = 47 if State=="TN"
> replace Statecode = 48 if State=="TX"
> replace Statecode = 49 if State=="UT"
> replace Statecode = 50 if State=="VT"
> replace Statecode = 51 if State=="VA"
> replace Statecode = 53 if State=="WA"
> replace Statecode = 54 if State=="WV"
> replace Statecode = 55 if State=="WI"
> replace Statecode = 56 if State=="WY"
> drop if Statecode==0
> gen FIPS = Statecode*1000 + Countycode
```

Now I need to keep the permits that were valid at least during a period of time from 2000 to 2010. Run the following command in order to do so.

```
> keep if (year(ExpirationDate)>2000 & year(ExpirationDate)<2011) | (year(IssuanceDate)>2000 &
year(IssuanceDate)<2011)
```

There are observations with FIPS information and observations with no FIPS data, thus I split the dataset into two. By dropping and keeping some observations, I ended up with two files: i) one for the observations with missing FIPS (`keep if missing(FIPS)|Countycode==0|Countycode==999`), and ii) one for the observations with FIPS (`drop if missing(FIPS)|Countycode==0|Countycode==999`). The observations with 0 or 999 as Countycode were considered as observations with missing FIPS.

Now, using the file with observations with FIPS, we can collapse it to count how many permits there are per county, by running the following command:

```
> collapse (count) Countycode , by(FIPS)
> rename Countycode NumberPermits
> label variable NumberPermits "Number of Permits"
```

Then, we can work on the observations with missing FIPS. First, I used the place-county relationship file in order to assign the corresponding county to permits that had state and city information. I did this by running a many-to-many merge. Once you merge these two datasets, you will obtain a dataset with county information, however, a single permit may have more than one county since the merge function assigned all the counties the city intersects. Therefore, it is important to assign only one county to a permit. In order to have the dataset in order I ran the following commands:

```
> sort statecode NPDES placename afact
> duplicates report NPDES
> edit
```

Then, a new window opens and then I started browsing through the dataset and dropping the duplicate records. The 'duplicates report NPDES' command will give you information regarding the number of NPDES duplicates in the dataset. You can keep using this command to verify that there are no more duplicate records. Also, it may be the case that the county assigned to the permit may not have the highest share of the population (see variable: `afact`). Therefore, it is important to pay attention to this variable and then change the FIPS code to the one with the highest share of the population when necessary.

After having dropped the duplicates and assigned the corresponding county to each one of the permits that had no FIPS information, now we can save this file and then keep the information that *now* has FIPS information. Then we can collapse the information in a similar fashion as performed for the records that did have FIPS information.

Then we can reopen the previously saved file and drop the information that still does not have FIPS information and then keep the records that latitude and longitude information. I dropped the observations with no latitude or longitude (`drop if missing(Latitude)|missing(Longitude)`) and then plotted the remaining records on top of the county boundaries<sup>5</sup>. Also, when plotting the discharge points you may realize that some of them are very far away from the contiguous U.S.; I discarded these points, since I assumed that the records in them might be faulty. After doing this, I spatially joined the polygons (counties) to the points (permits). It is important to join the polygons to the points, and not the either way around, in order to select the option 'is closest to it', since there may be some effluent discharge points really close to the shore but not inside the county boundary. I am assuming that the perception of people living in a county close to a discharge point will be somehow affected by it, thus I should not discard these discharge points from the analysis. Once this spatial join is performed, you should end up with a point shapefile that stores the information of the county closest to it. Then, export the attribute table as a txt file and then import into Stata.

Now that we have imported this file into Stata, we can collapse its information in order to obtain the number of permits per county by running the following commandas:

```
> rename fips FIPS
> replace FIPS = state_1*1000 + county_1
> collapse (count) county_1 , by(FIPS)
> rename county_1 NumberPermits
> label variable NumberPermits "Number of Permits"
```

Then, we can append these three files and then collapse the number of permits by FIPS in order to obtain the sum of permits:

```
> collapse (sum) NumberPermits , by(FIPS)
> label variable NumberPermits "Number of Active Permits from 2001 to 2010"
> sort FIPS
```

After doing so, we need to merge this file with the FIPS codes dta file in order to obtain the number of permits for all the counties within the contiguous U.S. Once these files are merged, we need to work on the FIPS codes that were not matched. For instance, I had the 12025 FIPS code, which used to be used for the Dade County, FL, until its name was changed to Miami-Dade, FL and then its FIPS code was changed to 12086. This FIPS code analysis needs to be done for all the non-matching codes. Here are some websites that are helpful for carrying out this analysis:

<http://www.itl.nist.gov/fipspubs/fip6-4.htm>

<http://wonder.cdc.gov/wonder/help/Census1970-2000.html>

[http://www.nrcs.usda.gov/wps/portal/nrcs/detail/national/technical/?cid=nrcs143\\_013710](http://www.nrcs.usda.gov/wps/portal/nrcs/detail/national/technical/?cid=nrcs143_013710)

### Notes for creating permit-day information

I created this variable since by only counting the number of active permits we might be weighting more heavily permits with short durations and more lightly permits with long durations. For instance, by simply counting the number of active permits, we might count a permit even though it was active one day during the whole decade of analysis and count another permit even though it was active throughout the same decade. It is clear that the latter would have more influence over the environmental context of the areas whereas the former would be hardly perceptible (though this would depend also on the magnitude of the discharge and the carrying capacity of the receiving body).

---

<sup>5</sup> The county boundaries shapefile was downloaded from the U.S. Census website.

[http://www.census.gov/geo/www/cob/cbf\\_counties.html](http://www.census.gov/geo/www/cob/cbf_counties.html)

[www2.census.gov/geo/tiger/GENZ2010/gz\\_2010\\_us\\_050\\_00\\_5m.zip](http://www2.census.gov/geo/tiger/GENZ2010/gz_2010_us_050_00_5m.zip)

Therefore, in order to create this new variable, I ran the following commands before collapsing the information:

```
> gen permitday = ExpirationDate-IssuanceDate
> replace permitday = mdy(12,31,2010)-IssuanceDate if ExpirationDate>mdy(12,31,2010)
> collapse (sum) permitday , by(FIPS)
> label variable permitday "Sum of Permit-day per county"
```

## 28. Landfill Waste

The EPA Toxics Release Inventory (TRI) Program has developed an application that allows the user to select, sort, and filter TRI data. The software can be downloaded from the following website:

<http://www.epa.gov/tri/tridotnet/index.html>

Once you download the TRI.NET software, you need to update its database by running the prompt Data\Get Data. Downloading all these files may take a while; I suggest downloading one at a time in order to make sure that you have successfully downloaded each one of them. Trying to download them all at once may take several hours.

Here are useful links to learn how to use the TRI.NET software:

TRI.NET Quick Start: <http://www.epa.gov/tri/tridotnet/quickstart.html>

TRI.NET User's Guide: <http://www.epa.gov/tri/tridotnet/guide.html>

TRI.NET Tutorial with examples: <http://www.epa.gov/tri/tridotnet/tutorial.html>

Once you have updated the database you can create tables of the years of interest. You will need to build queries for each one of the tables. You should select the following variables in order to create the table for year 2010.

Grouping variables: County, State, FIPS

Data Variables: Waste Quantities: Select All

Filtering variables: Year: 2010

Then, before running the query, select the Data: TRI 2010-2011: UP 2011. You could select 2000 and 1990 years too and then select the Data: 1987-2011: UP 2011 but your computer might crash, so I do not recommend doing this. However, if you are in 2023, go for it, and if your computer crashes, then you might want to consider replacing your computer.

Then, you can export the file to Excel by clicking on the Excel icon on the top bar. You can repeat this procedure for years 2000 and 1990. Remember to save the Excel files.

Note: As you may have already seen, there are two types of datasets: UP and NA. NA files are the files that were uploaded when the annual release of TRI data was released. UP files are the files that were updated (includes revisions, withdrawals, and late fillings) by facilities since the initial NA file was uploaded. Therefore, I recommend that you work with the UP datasets.

It is very important to verify the units of the data. You can open generate a report at the TRI Explorer website ([http://iaspub.epa.gov/triexplorer/tri\\_quantity.chemical](http://iaspub.epa.gov/triexplorer/tri_quantity.chemical)) and then compare the values to the values from the csv file created previously. If the values match, then the units of the csv file are the same as the units of the report generated from the TRI Explorer website. In this case, the values matched and the units in the TRI Explorer website were "pounds" so now we know that the units of the csv files are pounds.

Then, one can import the csv files into Stata using the prompt File\Import\Text data created by a spreadsheet. Then select the csv file and use the default options.

Once the file has been imported into Stata, you can begin creating new variables for all the variables imported since these imported variables are expressed in pounds and for the purpose of the research we need the variables expressed in metric tons. For instance, I ran the following commands to format the dataset and have all the quantities expressed in metric tons:

```
> rename totalwastemanaged8188 totalwaste
> rename totalproductionrelatedwastemanag totalprodwaste
> rename v6 quantreleased_on_off
> rename v7 onsite_disp
> rename v8 onsite_other_disp
> rename v9 offsite_disp
> rename v10 offsite_other_disp
> rename v12 onsite_energrecov
> rename v13 offsite_energrecov
> rename v15 onsite_recycled
> rename v16 offsite_recycled
> rename v18 onsite_treated
> rename v19 offsite_treated
> rename v20 disasterwaste
> replace totalwaste=totalwaste*0.00045359
> replace totalprodwaste=totalprodwaste*0.00045359
> replace quantreleased_on_off=quantreleased_on_off*0.00045359
> replace onsite_disp=onsite_disp*0.00045359
> replace onsite_other_disp=onsite_other_disp*0.00045359
> replace offsite_disp=offsite_disp*0.00045359
> replace offsite_other_disp=offsite_other_disp*0.00045359
> replace energyrecovery=energyrecovery*0.00045359
> replace onsite_energrecov=onsite_energrecov*0.00045359
> replace offsite_energrecov=offsite_energrecov*0.00045359
> replace recycled= recycled *0.00045359
> replace onsite_recycled=onsite_recycled*0.00045359
> replace offsite_recycled=offsite_recycled*0.00045359
> replace treated= treated*0.00045359
> replace onsite_treated=onsite_treated*0.00045359
> replace offsite_treated=offsite_treated*0.00045359
> replace disasterwaste=disasterwaste*0.00045359
```

Now that you have all the quantities in metric tonnes, you can save the dataset as a dta file. I also merged this dataset with the list of counties by FIPS code in order to get all the counties in the dta file, then replace the missing values with zeros and then run the ‘sum’ command in order to get the mean total waste managed at the county level.

```
> replace totalwaste=0 if missing(totalwaste)
> replace totalprodwaste=0 if missing(totalprodwaste)
> replace quantreleased_on_off=0 if missing(quantreleased_on_off)
> replace onsite_disp=0 if missing(onsite_disp)
> replace onsite_other_disp=0 if missing(onsite_other_disp)
> replace offsite_disp=0 if missing(offsite_disp)
> replace offsite_other_disp=0 if missing(offsite_other_disp)
> replace energyrecovery=0 if missing(energyrecovery)
> replace onsite_energrecov=0 if missing(onsite_energrecov)
> replace offsite_energrecov=0 if missing(offsite_energrecov)
> replace recycled=0 if missing(recycled)
> replace onsite_recycled=0 if missing(onsite_recycled)
> replace offsite_recycled=0 if missing(offsite_recycled)
> replace treated=0 if missing(treated)
> replace onsite_treated=0 if missing(onsite_treated)
> replace offsite_treated=0 if missing(offsite_treated)
> replace disasterwaste=0 if missing(disasterwaste)
```

I also dropped the observations that contained information from outside the contiguous U.S. by running the following command:

```
> drop if (state=="Alaska"|state=="Hawaii"|fips>56999)
```

## 29. Superfund

Superfund data can be accessed from <http://cumulis.epa.gov/supercpad/cursites/srchsites.cfm>. Data is retrieved in 2011 June. Raw data is stored in C:\Data\Superfund\superfund sites (NPL)\_20110624. Data is converted into STATA, stored in C:\Data\Superfund\ super.dta.

## 30. Treatment, storage and disposal facilities

See Large Quantity Generators of hazardous waste.

## 31. Large quantity generators of hazardous waste

The LQG and TSD data is presented every two years in the biennial reports. The 2009 Biennial Report can be downloaded from the following website: [ftp://ftp.epa.gov/rcrainfodata/br\\_2009/](ftp://ftp.epa.gov/rcrainfodata/br_2009/).

Once this data is downloaded, it needs to be unzipped twice. The file that contains the LQG and TSD information is the s1.txt file. In order to create the dictionary file, it is important to use the 'File specification guide'. The links for the 1999 and 2009 guides are the following:

<http://www.epa.gov/epawaste/inforesources/data/brs99/brshelp.pdf>

<http://www.epa.gov/epawaste/inforesources/data/br09/br09-specification.pdf>

In order to import the file for year 2009 a dictionary file was created with the following syntax:

```
infile dictionary {
_column(1) str12 H_ID "EPA Identification Number"
_column(13) str8 Date "Received Date"
_column(21) str80 H_Name "Handler Name"
_column(101) str12 Locat1 "Location Street Number"
_column(113) str30 Locat2 "Location Street 1"
_column(143) str30 Locat3 "Location Street 2"
_column(173) str25 Locat4 "Location City"
_column(198) str2 Locat5 "Location State"
_column(200) str14 Locat6 "Location Zip"
_column(214) str5 Locat7 "Location County Code"
_column(219) str2 Locat8 "Location Country Code"
_column(221) str10 Locat9 "State District"
_column(231) str12 Mail1 "Mailing Street Number"
_column(243) str30 Mail2 "Mailing Street 1"
_column(273) str30 Mail3 "Mailing Street 2"
_column(303) str25 Mail4 "Mailing City"
_column(328) str2 Mail5 "Mailing State"
_column(330) str14 Mail6 "Mailing Zip"
_column(344) str2 Mail7 "Mailing Country Code"
_column(346) str1 SiteLand "Site Land Type"
_column(347) str15 Con1 "Contact First Name"
_column(362) str1 Con2 "Contact Middle Initial"
_column(363) str15 Con3 "Contact Last Name"
_column(378) str12 Con4 "Contact Street Number"
_column(390) str30 Con5 "Contact Street 1"
_column(420) str30 Con6 "Contact Street 2"
_column(450) str25 Con7 "Contact City"
_column(475) str2 Con8 "Contact State"
_column(477) str14 Con9 "Contact Zip"
_column(491) str2 Con10 "Contact Country"
_column(493) str10 Con11 "Contact Phone Number"
_column(503) str6 Con12 "Contact Phone Number Extension"
_column(509) str15 Con13 "Contact Fax Number"
_column(524) str45 Con14 "Contact Title"
_column(569) str80 Con15 "Contact E-mail Address"
```

```
_column(649) str1 FedGen "Federal Generator Status"
_column(650) str1 StaGen "State Generator Status"
_column(651) str1 ShortGen "Short Term or Temporary Generator"
_column(652) str1 Importer "U.S. Importer of Hazardous Waste"
_column(653) str1 Mixed "Mixed Waste (hazardous and radioactive) Generator"
_column(654) str1 Trans "Transporter of Hazardous Waste"
_column(655) str1 Transfer "Transfer Facility of Hazardous Waste"
_column(656) str1 TSDwaste "Treater, Storer, or Disposer of Hazardous Waste in a Permitted Unit"
_column(657) str1 Recycler "Recycler of Hazardous Waste"
_column(658) str1 SmallQua "Small Quantity On-Site Burner Exemption"
_column(659) str1 Smelting "Smelting, Melting, and Refining Furnace Exemption"
_column(660) str1 Undergro "Underground Injection Control"
_column(661) str1 Offsite "Received Hazardous Waste from Offsite"
_column(662) str1 Univ "Destination Facility for Universal Waste"
_column(663) str1 UsedOil1 "Used Oil Transporter"
_column(664) str1 UsedOil2 "Used Oil Transfer Facility"
_column(665) str1 UsedOil3 "Used Oil Processor"
_column(666) str1 UsedOil4 "Used Oil Re-refiner"
_column(667) str1 UsedOil5 "Off-Specification Used Oil Burner"
_column(668) str1 Market1 "Marketer Who Directs Shipment of Off-Specification Used Oil to Off-Specification Used Oil Burner"
_column(669) str1 Market2 "Marketer Who First Claims the Used Oil Meets the Specifications"
_column(670) str1 Opting1 "Opting into or Currently Operating under 40 CFR Part 262 Subpart K as a College or University"
_column(671) str1 Opting2 "Opting into or Currently Operating under 40 CFR Part 262 Subpart K as a Teaching Hospital"
_column(672) str1 Opting3 "Opting into or Currently Operating under 40 CFR Part 262 Subpart K as a Non-profit Research Institute"
_column(673) str1 Opting4 "Withdrawing from 40 CFR Part 262 Subpart K"
_column(674) str1 NHWR "Include this Information in the National Hazardous Waste Report"
_column(675) str100 Comm "Comments and Notes"

}
```

Now, even though the dictionary file allows defining variables as numeric, I have found it better to import all the variables as string and then ‘destring’ them if necessary. For instance, I tried to import the s1.txt file defining date as numeric but kept getting error messages. When defining data as string, the file was imported with no error messages.

Now that the file has been imported as a dta file one needs to create Boolean variables that collect information regarding the ‘Large quantity generator’ status and the ‘Treatment, storage and disposal’ status of the facilities. In order to do this, one needs to run the following commands:

```
> gen TSD = 0
> replace TSD = 1 if TSDwaste=="Y"
> gen LQG = 0
> replace LQG = 1 if FedGen=="1"
> keep Locat5 Locat7 TSD LQG
> drop if missing(Locat5) | missing(Locat7)
> rename Locat5 state
> drop if state=="AK"|state=="HI"|state=="PR"|state=="VI"
> generate countycode = substr(Locat7,3,5)
> desting countycode , replace
> generate statecode = 0
> replace statecode = 1 if state=="AL"
> replace statecode = 4 if state=="AZ"
> replace statecode = 5 if state=="AR"
> replace statecode = 6 if state=="CA"
> replace statecode = 8 if state=="CO"
> replace statecode = 9 if state=="CT"
> replace statecode = 10 if state=="DE"
> replace statecode = 11 if state=="DC"
> replace statecode = 12 if state=="FL"
> replace statecode = 13 if state=="GA"
> replace statecode = 16 if state=="ID"
> replace statecode = 17 if state=="IL"
> replace statecode = 18 if state=="IN"
> replace statecode = 19 if state=="IA"
> replace statecode = 20 if state=="KS"
```



```
> replace statecode = 21 if state=="KY"
> replace statecode = 22 if state=="LA"
> replace statecode = 23 if state=="ME"
> replace statecode = 24 if state=="MD"
> replace statecode = 25 if state=="MA"
> replace statecode = 26 if state=="MI"
> replace statecode = 27 if state=="MN"
> replace statecode = 28 if state=="MS"
> replace statecode = 29 if state=="MO"
> replace statecode = 30 if state=="MT"
> replace statecode = 31 if state=="NE"
> replace statecode = 32 if state=="NV"
> replace statecode = 33 if state=="NH"
> replace statecode = 34 if state=="NJ"
> replace statecode = 35 if state=="NM"
> replace statecode = 36 if state=="NY"
> replace statecode = 37 if state=="NC"
> replace statecode = 38 if state=="ND"
> replace statecode = 39 if state=="OH"
> replace statecode = 40 if state=="OK"
> replace statecode = 41 if state=="OR"
> replace statecode = 42 if state=="PA"
> replace statecode = 44 if state=="RI"
> replace statecode = 45 if state=="SC"
> replace statecode = 46 if state=="SD"
> replace statecode = 47 if state=="TN"
> replace statecode = 48 if state=="TX"
> replace statecode = 49 if state=="UT"
> replace statecode = 50 if state=="VT"
> replace statecode = 51 if state=="VA"
> replace statecode = 53 if state=="WA"
> replace statecode = 54 if state=="WV"
> replace statecode = 55 if state=="WI"
> replace statecode = 56 if state=="WY"
> drop if statecode==0
> gen FIPS = statecode*1000+countycode
> order FIPS , first
> collapse (sum) TSD LQG , by(FIPS)
> label variable TSD "Number of Treatment, storage and disposal facilities"
> label variable LQG "Number of Large Quantity Generators"
```

Now, there are some FIPS codes that are not consistent with the latest codes. For instance, Dade County, Florida, was renamed Miami-Dade County, Florida, changing its FIPS code from 12025 to 12086. Therefore, we need to run the following commands:

```
> replace FIPS = 12086 if FIPS==12025
```

Now we can save the file, and this file will contain three fields. The first field has the FIPS code, the second one has the number of TSD facilities and the third one has the number of LQGs. After doing this, we can merge this file with the FIPS dta file and then replace the missing values from the counties that have neither TSD nor LQG by running the following commands:

```
> replace TSD=0 if missing(TSD)
> replace LQG=0 if missing(LQG)
```

## 32. Nuclear power plants

Nuclear power plants info is collected from <http://www.nrc.gov/info-finder/reactor/index.html#listAlpha> into excel manually. Also, data from decommissioned nuclear power plants was collected from the following website: <http://www.nrc.gov/info-finder/decommissioning/>.

The fields created in the Excel file are the following:

Plant: name of the plant

Unit: Number of the unit. Could be 1, 2, 3, depending on the number of nuclear units

Type: 'Boiling Water Reactor' or 'Pressurized water reactor'

Location: Address or reference of location.

county: county

state: state

OperatingLicenseIssued: Date when license was issued

LicenseExpires: Date when license expires

LicensedMWt: Licensed MWt

MWelectrical: MWe

OwnerOperator: Owner/Operator

website1: website where information was found

website2: website where information was found

website3: website where information was found

After preparing the excel file, we need to import it into Stata. Once this file has been imported, we can run the following commands in order to finally have a dta file that summarized the number of nuclear plants, number of nuclear reactors and MWe per county.

```
> keep if year(OperatingLicenseIssued)<2010&year(LicenseExpires)>2010
> replace county=proper(county)
> merge m:1 county state using "E:\Research\FIPS\FIPS.dta"
> drop if missing(Plant)
> replace FIPS=17039 if county=="Dewitt"&state=="Illinois"
> collapse (count) Unit (sum) MWelectrical , by(Plant FIPS)
> gen nuclearplant = 1
> rename Unit unit
> collapse (sum) nuclearplant unit MWelectrical , by (FIPS)
> rename unit nuclearreactor
> label variable nuclearplant "Number of nuclear plants"
> label variable nuclearreactor "Number of nuclear reactors"
> label variable MWelectrical "Total MWelectrical produced"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> replace nuclearplant = 0 if missing(nuclearplant)
> replace nuclearreactor = 0 if missing(nuclearreactor)
> replace MWelectrical = 0 if missing(MWelectrical)
> drop statecode
> drop _merge
> save "E:\Research\Nuclear\Nuclear2010_collapse.dta"
```

The same procedure can be followed to create the 2000, 1990 and 1980 datasets.

### 33. Coal-fired power plants

Coal-fired power plants info is collected from the Annual Electric Generator data website:

<http://www.eia.gov/electricity/data/eia860>. We can download data per year on this website. According to Davis (2011)<sup>6</sup> plant construction typically takes at least two years. Thus, assuming that households internalize the presence of a power plant when the plant enters construction phase, plants expected to be ready by 2011 must have had effects on the real estate market by 2009. Also, since the 2010 Census, which gathers information throughout 2009, is going to be an important input for this assessment, I decided to use information from the 2009 plants, rather than 2010. Therefore, I used information from plants that were operating throughout 2009 and plants that by December 2009 were proposed to be in operation by December 2011.

Once the 2009 data was downloaded from the website, we can unzip the file and we'll find several Excel files and a pdf file. From these Excel files, the one we need is the one named 'GeneratorY09.xlsx'. From this file, we

<sup>6</sup> Lucas W. Davis 2011. *The Effect of Power Plants on Local Housing Values and Rents*. The Review of Economics and Statistics 2011, 93(4): 1391-1402.

need the worksheets 'Exist' and 'Prop', which have information of existing power plants and proposed power plants, respectively. We need to import these two worksheets and format them keeping the fields and observations of interest for the purpose of the research. Run the following commands:

```
> import excel "E:\Research\EIA-860\2009\GeneratorY09.xls", sheet("Exist") firstrow
> keep PLANT_CODE PLANT_NAME STATE COUNTY GENERATOR_ID STATUS NAMEPLATE ENERGY_SOURCE_1
> rename PLANT_CODE plant_code
> rename PLANT_NAME plant_name
> rename STATE state
> rename COUNTY county
> rename GENERATOR_ID generator_id
> rename STATUS status
> rename NAMEPLATE MW
> rename ENERGY_SOURCE_1 primarysource
> keep if primarysource=="BIT"|primarysource=="LIG"|primarysource=="SC"|primarysource=="SUB"|primarysource=="WC"
> keep if status=="OP"
> gen generator100MW = 0
> replace generator100MW = 1 if MW>=100
> gen MW_gen100MW = 0
> replace MW_gen100MW = MW if generator100MW == 1
> gen generator = 1
> collapse (sum) generator generator100MW MW MW_gen100MW , by(plant_code plant_name state county)
> gen plant = 1
> gen plant100MW = 0
> replace plant100MW = 1 if MW>=100
> gen plant_gen100MW = 0
> replace plant_gen100MW = 1 if generator100MW>0
> gen generator_plant100MW = 0
> replace generator_plant100MW = generator if MW>=100
> gen MW_plant100MW = 0
> replace MW_plant100MW = MW if MW>=100
> collapse (sum) plant plant100MW plant_gen100MW generator generator_plant100MW generator100MW MW MW_plant100MW
MW_gen100MW , by(state county)
> label variable plant "Number of coal-fired plants"
> label variable plant100MW "Number of coal-fired plants with power >= 100MW"
> label variable plant_gen100MW "Number of coal-fired plants with at least 1 generator with power >= 100MW"
> label variable generator "Number of generators"
> label variable generator_plant100MW "Number of generators in plants with combined power >= 100MW"
> label variable generator100MW "Number of generators with power >= 100MW"
> label variable MW "Total MW"
> label variable MW_plant100MW "Total MW from plants with power >= 100MW"
> label variable MW_gen100MW "Total MW from generators with power >= 100MW"
> save "E:\Research\EIA-860\2009\CoalPlant_exist_collapse.dta"

> import excel "E:\Research\EIA-860\2009\GeneratorY09.xls", sheet("Prop") firstrow clear
> drop if CURRENT_YEAR>2011
> keep PLANT_CODE PLANT_NAME STATE COUNTY GENERATOR_ID STATUS NAMEPLATE ENERGY_SOURCE_1
> rename PLANT_CODE plant_code
> rename PLANT_NAME plant_name
> rename STATE state
> rename COUNTY county
> rename GENERATOR_ID generator_id
> rename STATUS status
> rename NAMEPLATE MW
> rename ENERGY_SOURCE_1 primarysource
> keep if primarysource=="BIT"|primarysource=="LIG"|primarysource=="SC"|primarysource=="SUB"|primarysource=="WC"
> drop if status=="IP"|status=="P"
> gen generator100MW = 0
> replace generator100MW = 1 if MW>=100
> gen MW_gen100MW = 0
> replace MW_gen100MW = MW if generator100MW == 1
> gen generator = 1
> collapse (sum) generator generator100MW MW MW_gen100MW , by(plant_code plant_name state county)
> gen plant = 1
> gen plant100MW = 0
> replace plant100MW = 1 if MW>=100
> gen plant_gen100MW = 0
> replace plant_gen100MW = 1 if generator100MW>0
> gen generator_plant100MW = 0
> replace generator_plant100MW = generator if MW>=100
> gen MW_plant100MW = 0
> replace MW_plant100MW = MW if MW>=100
```

```
> collapse (sum) plant plant100MW plant_gen100MW generator generator_plant100MW generator100MW MW MW_plant100MW
MW_gen100MW , by(state county)
> label variable plant "Number of coal-fired plants"
> label variable plant100MW "Number of coal-fired plants with power >= 100MW"
> label variable plant_gen100MW "Number of coal-fired plants with at least 1 generator with power >= 100MW"
> label variable generator "Number of generators"
> label variable generator_plant100MW "Number of generators in plants with combined power >= 100MW"
> label variable generator100MW "Number of generators with power >= 100MW"
> label variable MW "Total MW"
> label variable MW_plant100MW "Total MW from plants with power >= 100MW"
> label variable MW_gen100MW "Total MW from generators with power >= 100MW"
> save "E:\Research\EIA-860\2009\CoalPlant_prop_collapse.dta"
```

Now that we have both files, one for the existing plants and one for the proposed plants, we can merge them together.

```
> use "E:\Research\EIA-860\2009\CoalPlant_exist_collapse.dta", clear
> append using "E:\Research\EIA-860\2009\CoalPlant_prop_collapse.dta"
> collapse (sum) plant plant100MW plant_gen100MW generator generator_plant100MW generator100MW MW MW_plant100MW
MW_gen100MW , by(state county)
> sort state county
> gen statecode = 0
> label variable statecode "State code"
> replace statecode = 1 if state=="AL"
> replace statecode = 4 if state=="AZ"
> replace statecode = 5 if state=="AR"
> replace statecode = 6 if state=="CA"
> replace statecode = 8 if state=="CO"
> replace statecode = 9 if state=="CT"
> replace statecode = 10 if state=="DE"
> replace statecode = 11 if state=="DC"
> replace statecode = 12 if state=="FL"
> replace statecode = 13 if state=="GA"
> replace statecode = 16 if state=="ID"
> replace statecode = 17 if state=="IL"
> replace statecode = 18 if state=="IN"
> replace statecode = 19 if state=="IA"
> replace statecode = 20 if state=="KS"
> replace statecode = 21 if state=="KY"
> replace statecode = 22 if state=="LA"
> replace statecode = 23 if state=="ME"
> replace statecode = 24 if state=="MD"
> replace statecode = 25 if state=="MA"
> replace statecode = 26 if state=="MI"
> replace statecode = 27 if state=="MN"
> replace statecode = 28 if state=="MS"
> replace statecode = 29 if state=="MO"
> replace statecode = 30 if state=="MT"
> replace statecode = 31 if state=="NE"
> replace statecode = 32 if state=="NV"
> replace statecode = 33 if state=="NH"
> replace statecode = 34 if state=="NJ"
> replace statecode = 35 if state=="NM"
> replace statecode = 36 if state=="NY"
> replace statecode = 37 if state=="NC"
> replace statecode = 38 if state=="ND"
> replace statecode = 39 if state=="OH"
> replace statecode = 40 if state=="OK"
> replace statecode = 41 if state=="OR"
> replace statecode = 42 if state=="PA"
> replace statecode = 44 if state=="RI"
> replace statecode = 45 if state=="SC"
> replace statecode = 46 if state=="SD"
> replace statecode = 47 if state=="TN"
> replace statecode = 48 if state=="TX"
> replace statecode = 49 if state=="UT"
> replace statecode = 50 if state=="VT"
> replace statecode = 51 if state=="VA"
> replace statecode = 53 if state=="WA"
> replace statecode = 54 if state=="WV"
> replace statecode = 55 if state=="WI"
> replace statecode = 56 if state=="WY"
```

```
> drop if statecode==0
> merge 1:1 county statecode using "E:\Research\FIPS\FIPS.dta"
> drop if missing(plant)
> replace FIPS = 18091 if state=="IN"&county=="La Porte"
> replace FIPS = 18141 if state=="IN"&county=="St Joseph"
> replace FIPS = 21145 if state=="KY"&county=="McCracken"
> replace FIPS = 24033 if state=="MD"&county=="Prince Georges"
> replace FIPS = 26147 if state=="MI"&county=="St Clair"
> replace FIPS = 27137 if state=="MN"&county=="St Louis"
> replace FIPS = 29183 if state=="MO"&county=="St Charles"
> replace FIPS = 29189 if state=="MO"&county=="St Louis"
> replace FIPS = 29510 if state=="MO"&county=="St Louis City"
> replace FIPS = 38055 if state=="ND"&county=="McLean"
> replace FIPS = 35031 if state=="NM"&county=="McKinley"
> replace FIPS = 40079 if state=="OK"&county=="Leflore"
> replace FIPS = 51510 if state=="VA"&county=="Alexandria"
> replace FIPS = 51550 if state=="VA"&county=="Chesapeake"
> replace FIPS = 51670 if state=="VA"&county=="City of Hopewell"
> replace FIPS = 51760 if state=="VA"&county=="City of Richmond"
> sort FIPS
> collapse (sum) plant plant100MW plant_gen100MW generator generator_plant100MW generator100MW MW MW_plant100MW
MW_gen100MW , by(FIPS)
> label variable plant "Number of coal-fired plants"
> label variable plant100MW "Number of coal-fired plants with power >= 100MW"
> label variable plant_gen100MW "Number of coal-fired plants with at least 1 generator with power >= 100MW"
> label variable generator "Number of generators"
> label variable generator_plant100MW "Number of generators in plants with combined power >= 100MW"
> label variable generator100MW "Number of generators with power >= 100MW"
> label variable MW "Total MW"
> label variable MW_plant100MW "Total MW from plants with power >= 100MW"
> label variable MW_gen100MW "Total MW from generators with power >= 100MW"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop _merge statecode
> sort FIPS
> replace plant = 0 if missing(plant)
> replace plant100MW = 0 if missing(plant100MW)
> replace plant_gen100MW = 0 if missing(plant_gen100MW)
> replace generator = 0 if missing(generator)
> replace generator_plant100MW = 0 if missing(generator_plant100MW)
> replace generator100MW = 0 if missing(generator100MW)
> replace MW = 0 if missing(MW)
> replace MW_plant100MW = 0 if missing(MW_plant100MW)
> replace MW_gen100MW = 0 if missing(MW_gen100MW)
> save "E:\Research\EIA-860\2009\CoalPlant_collapse.dta"
```

## 34. PM2.5

EPA has a dataset of air quality concentrations estimation using downscaling for PM2.5 and O3, for years 2001 to 2008. The webpage is the following:

[http://www.epa.gov/nerlesd1/land-sci/lcb/lcb\\_faqs.html](http://www.epa.gov/nerlesd1/land-sci/lcb/lcb_faqs.html)

However, 2001 data is not available for the whole Contiguous U.S., this data is only available for the eastern part of the U.S.

### 2008 Downscaling model

The data is downloaded in .csv format. After importing into Stata, there will be 6 variables. According to the "DSMetadata\_Air" file downloaded from that website, the variables can be renamed in the following manner:

```
> insheet using "E:\Research\EPA-AirQuality\PM2.5\data\Downscaling\Predictions08PM-Census2010edited.csv", clear
> rename v1 date, replace
> rename v2 FIPS, replace
> rename v3 latitude, replace
> rename v4 longitude, replace
> rename v5 PM25, replace
> rename v6 std_error, replace
```

To obtain the annual mean concentration, you can run the following commands:

```
> collapse PM25 , by(FIPS)
> label variable PM25 "Mean annual PM2.5 concentration in ug/m3"
> label variable FIPS "Census tract FIPS code"
> merge 1:1 FIPS using "E:\Research\Population\pop_censustract2010.dta"
> drop if State==2|State==15|State>56
> sum Population if(_merge==2)
```

If Population is equal to 0 in all the observations with `_merge==2`, then we can drop them, since they will not affect the calculations. Then we can calculate the population-weighted mean annual PM2.5 concentration at the county level.

```
> keep if _merge==3
> keep PM25 State County Population
> gen FIPS = State*1000+County
> drop State County
> gen popPM25 = Population*PM25
> collapse (sum) popPM25 Population , by(FIPS)
> gen PM25 = popPM25/Population
> keep FIPS PM25
> label variable PM25 "Mean annual PM2.5 concentration in ug/m3"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> save "E:\Research\EPA-AirQuality\PM2.5\PM2.5_2008.dta"
```

## 2001 Downscaling model

Working with the 2001 downscaling model is similar to working with the 2008 downscaling model. However, we would not obtain results for the whole Contiguous U.S. Also, results are provided for the 2000 blockgroups, so we cannot use them to collapse data to the county level, since we would be calculating data for the 2000 counties, not 2010 counties.

Since we do not have information for all the census tracts within the Contiguous U.S., first we need to create a dta file that has a dummy variable that says if we have information for all the census tracts within a given county or not.

```
> insheet using "E:\Research\EPA-AirQuality\PM2.5\data\Downscaling\Predictions01PM-Census2000edited.csv", clear
> rename v1 date, replace
> rename v2 FIPS, replace
> rename v3 latitude, replace
> rename v4 longitude, replace
> rename v5 PM25, replace
> rename v6 std_error, replace
> collapse PM25 , by(FIPS)
> label variable PM25 "Mean annual PM2.5 concentration in ug/m3"
> label variable FIPS "Census tract FIPS code"
> merge 1:1 FIPS using "E:\Research\Population\pop_censustract2000.dta"
> drop if _merge==1
> keep FIPS PM25 Population STATE
> destring , replace
> drop if STATE==2|STATE==15|STATE>56
> sort FIPS
> gen miss = 1 if missing(PM25)
> replace miss = 0 if missing(miss)
> gen FIPScounty = floor(FIPS/1000000)
> collapse (sum) miss , by(FIPScounty)
> browse
> gen keep = 0
> label variable keep "1 if county has PM2.5 data for every tract, 0 if not"
> rename FIPScounty FIPS
> replace keep = 1 if miss==0
> keep FIPS keep
> save "E:\Research\EPA-AirQuality\PM2.5\PM2.5_2001_keep.dta"
```

```
> insheet using "E:\Research\EPA-AirQuality\PM2.5\data\Downscaling\Predictions01PM-Census2000edited.csv", clear
> rename v1 date, replace
> rename v2 FIPS, replace
> rename v3 latitude, replace
> rename v4 longitude, replace
> rename v5 PM25, replace
> rename v6 std_error, replace
> collapse PM25 , by(FIPS)
> label variable PM25 "Mean annual PM2.5 concentration in ug/m3"
> label variable FIPS "Census tract FIPS code"
> merge 1:1 FIPS using "E:\Research\Population\pop_censustract2000.dta"
> drop if _merge==1
> keep FIPS PM25 Population STATE
> destring , replace
> drop if STATE==2|STATE==15|STATE>56
> sort FIPS
> gen index = Population*PM25
> rename FIPS FIPStract
> gen FIPS = floor(FIPStract/1000000)
> label variable FIPS "County FIPS code"
> collapse (sum) Population index , by(FIPS)
> gen PM25 = index/Population
> label variable PM25 "Mean annual PM2.5 concentration in ug/m3"
> merge 1:1 FIPS using "E:\Research\EPA-AirQuality\PM2.5\PM2.5_2001_keep.dta"
> keep if keep==1
> keep FIPS PM25
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if missing(PM25)
> save "E:\Research\EPA-AirQuality\PM2.5\PM2.5_2001_downscaling.dta"
```

This dta file contains PM2.5 mean annual concentrations for counties which have information for all of their corresponding census tracts. If one county had at least one census tract with no information, the concentration was not collapsed for this county. However, keep in mind that we still have no information for some counties.

## 2000 Detailed Air Quality System Data

Since we need to estimate PM2.5 concentrations for the counties that did not have complete information in the downscaling model, we need to plot the PM2.5 concentrations gathered from the Detailed Air Quality System Data. A detailed description of how we can work with this data is provided in Section 35 PM10. I will only write down the commands used to format the data.

```
> insheet using "E:\Research\EPA-AirQuality\PM2.5\data\AQ5\RD_501_88101_2000-0.txt", delimiter(",") clear
> tab rd
> keep if rd == "RD"
> tab actioncode
> tab parameter
> drop rd actioncode parameter
> rename samplevalue PM25
> split starttime, parse(:)
> drop starttime
> rename starttime1 hour
> rename starttime2 minute
> destring hour minute , replace
> drop if (statecode=="CC")
> destring , replace
> drop if (statecode==2|statecode==15|statecode>56)
> destring , replace
> gen year = floor(date/10000)
> gen month = floor((date-year*10000)/100)
> gen day = date-year*10000-month*100
> keep if missing(qualifier1) | (substr(qualifier1,1,1)=="I")
> keep if missing(qualifier2) | (substr(qualifier2,1,1)=="I")
> keep if missing(qualifier3) | (substr(qualifier3,1,1)=="I")
> keep if missing(qualifier4) | (substr(qualifier4,1,1)=="I")
> keep if missing(qualifier5) | (substr(qualifier5,1,1)=="I")
> keep if missing(qualifier6) | (substr(qualifier6,1,1)=="I")
> keep if missing(qualifier7) | (substr(qualifier7,1,1)=="I")
> keep if missing(qualifier8) | (substr(qualifier8,1,1)=="I")
> keep if missing(qualifier9) | (substr(qualifier9,1,1)=="I")
```

```
> keep if missing(qualifier10) | (substr(qualifier10,1,1)=="I")
> drop qualifier1 qualifier2 qualifier3 qualifier4 qualifier5 qualifier6 qualifier7 qualifier8 qualifier9
qualifier10
> tab unit
> drop if missing(PM25)
> replace PM25 = PM25*1000 if unit=="5"
> drop unit
> label variable PM25 "PM2.5 in ug/m3"
> replace samplingfrequency=0 if missing(samplingfrequency)
> tab samplingfrequency sampleduration
> collapse (mean) PM25 , by(statecode countycode siteid sampleduration samplingfrequency year month day hour)
> collapse (mean) PM25 (count) hour , by(statecode countycode siteid sampleduration samplingfrequency year month
day)
> tab hour
> drop if (hour<18&sampleduration==1)
> gen byte boolean_sf0 = samplingfrequency==0
> gen byte boolean_sf1 = samplingfrequency==1
> gen byte boolean_sf2 = samplingfrequency==2
> gen byte boolean_sf3 = samplingfrequency==3
> gen byte boolean_sf6 = samplingfrequency==6
> gen byte boolean_sf7 = samplingfrequency==7
> gen byte boolean_sf9 = samplingfrequency==9
> gen byte boolean_sf11 = samplingfrequency==11
> gen byte boolean_sd1 = sampleduration==1
> gen byte boolean_sd7 = sampleduration==7
> collapse (mean) PM25 (sum) boolean_sf0 boolean_sf1 boolean_sf2 boolean_sf3 boolean_sf6 boolean_sf7 boolean_sf9
boolean_sf11 boolean_sd1 boolean_sd7 , by(statecode countycode siteid year month day)
> collapse (mean) PM25 (sum) boolean_sf0 boolean_sf1 boolean_sf2 boolean_sf3 boolean_sf6 boolean_sf7 boolean_sf9
boolean_sf11 boolean_sd1 boolean_sd7 , by(statecode countycode siteid year month)
> keep if (((boolean_sd1>=21 | boolean_sf1>=21 | boolean_sf2>=10 | boolean_sf3>=7 | boolean_sf6>=3 |
boolean_sf7>=2)&month==2) | ((boolean_sd1>=22 | boolean_sf1>=22 | boolean_sf2>=11 | boolean_sf3>=7 |
boolean_sf6>=3 | boolean_sf7>=2)&(month==4 | month==6 | month==9 | month==11)) | ((boolean_sd1>=23 |
boolean_sf1>=23 | boolean_sf2>=12 | boolean_sf3>=8 | boolean_sf6>=4 | boolean_sf7>=2)&(month==1 | month==3 |
month==5 | month==7 | month==8 | month==10 | month==12)))
> collapse (mean) PM25 (count) month , by(statecode countycode siteid year)
> drop if month<9
> collapse (mean) PM25 , by(statecode countycode siteid)
> label variable PM25 "Mean annual PM2.5 concentration in ug/m3"
> gen FIPS = statecode*1000+countycode
> order FIPS , first
> drop statecode countycode
> save "E:\Research\EPA-AirQuality\PM2.5\PM2.5_2000.dta"

> use "E:\Research\EPA-AirQuality\PM2.5\PM2.5_2000.dta", clear
> merge 1:1 FIPS siteid using "E:\Research\EPA-AirQuality\station_location\sites.dta"
> drop if _merge==2
> replace latitude = 0 if missing(latitude)
> replace longitude = 0 if missing(longitude)
> collapse (mean) PM25 , by(FIPS latitude longitude)
> rename latitude lat_station
> rename longitude long_station
> merge m:1 FIPS using "E:\Research\Population\PopCentroidCounty\County_Popcentroid.dta"
> keep if _merge==3
> replace lat_station = latitude if lat_station==0
> replace long_station = longitude if long_station==0
> keep FIPS lat_station long_station PM10
> rename lat_station latitude
> rename long_station longitude
> label variable PM10 "Mean annual PM10 concentration in ug/m3"
> sort FIPS
> save "E:\Research\EPA-AirQuality\PM10\PM10_2010_geo.dta"
> export excel using "E:\Research\EPA-AirQuality\PM10\PM10_2010_geo.xls", firstrow(variables)
```

Finally, see final section of Mean Precipitation (Section 2) to import the xls file into ArcGIS, plot it spatially and obtain the mean annual PM2.5 concentration at the county population-weighted centroids. However, since we have information from the downscaling model, we can use these two datasets together and use the one that we consider the most suitable.

Concentrations may be compared with the National Ambient Air Quality Standards  
<http://www.epa.gov/air/criteria.html#2>.



Air Quality Index may be used as well.

Definition: <http://airnow.gov/index.cfm?action=aqibasics.aqi>

Source (1970 - 2012): <http://www.epa.gov/ttn/airs/airsaqs/detaildata/AQIindex.htm>

For future references and comparisons, air quality concentrations can be downloaded directly from the following webpage:

For 2000: <http://www.epa.gov/airtrends/reports.html>

For 2010: <http://www.epa.gov/airtrends/factbook.html>

By downloading the air quality statistics by county for both years, we can make comparisons directly with the NAAQS and we make sure that the values used in the analyses correspond to the EPA values.

## 35. PM10

There is air quality data from 1993 to 2012, available for download from:

<http://www.epa.gov/ttn/airs/airsaqs/detaildata/downloadaqsdata.htm>. Data was downloaded for years 2000 and 2010 for the following parameters:

- PM2.5,
- PM10,
- Ozone,
- Sulphur dioxide,
- Carbon monoxide, and
- Nitrogen dioxide.

The following steps describe how to work with the PM10 data. PM2.5 and Ozone data from this website was not used to create the 2010 annual means, since there is a more appropriate model: Downscaling model. Working with Sulphur dioxide, Carbon monoxide and Nitrogen dioxide data is very similar to working with PM10, though it will be addressed in the respective sections to avoid confusion.

Data is available in concentrations reported in some cases at an hourly frequency and in other cases at frequencies ranging from 1 to 7 days, in txt format. In order to import the data into Stata, it was necessary to import it using the prompt File/Import/Text data created by the spreadsheet. However, before importing, one needs to make sure that the first line in the .txt file names each variable. If this is not the case, then editing the .txt file and naming each column is crucial. Open the txt file and identify the character used to separate columns. In this case, the character was '|'; it is always crucial to identify this character.

After selecting the option "File/Import/Text data created by the spreadsheet", a window pops up. In this window, we have to load the .txt file, and in the delimiter section, we have to select "User-specified delimiter" and type | in the box just below it.

```
> insheet using "E:\Research\EPA-AirQuality\PM10\data\AQS\RD_501_81102_2010-0.txt", delimiter("|") clear
```

After importing the file, we can get rid of the irrelevant variables. To erase these variables, we can run the 'drop' command (e.g., drop rd). I dropped the variables that had the same value for each observation, such as 'rd', and variables that had no information, such as qualifier4. To have a quick idea of the observations stored in each variable, one can run the commands 'sum \*variable name\*' or 'tab \*variable name'. After viewing the variables and the data they stored, I dropped some variables by running the drop command.

```
> tab rd
> keep if rd=="RD"
> tab actioncode
> tab parameter
> drop rd actioncode parameter
> rename samplevalue PM10
```

The start time variable may have been stored as a string variable; if this is the case, then we can run the following commands to convert it into a numeric variable.

```
> split starttime, parse(:)
> drop starttime
> rename starttime1 hour
> rename starttime2 minute
> destring hour , replace
> destring minute , replace
```

In some instances, the state code will be a string variable because it has some observations recorded as “CC”. When this is the case, we need to drop the observations in which the state code is equal to CC. Run the following command:

```
> drop if(statecode=="CC")
```

Then, destring the statecode variable by running the following command:

```
> destring statecode, replace
```

Since we are only analyzing the counties for the conterminous US, we need to drop all the observations whose statecode correspond to Alaska, Hawaii, Puerto Rico, Virgin Islands, and another area with FIPS code equal to 80; run the following command:

```
> drop if(statecode==2|statecode==15|statecode>56)
```

### Obtaining means and running the quality control process

To create the means first we need to create year, month and day variables. To do this we can run the following commands, assuming that the variable “date” stores the year, month and day information with the first four characters for the year, the next two for month and the next two for day:

```
> gen year = floor(date/10000)
> gen month = floor((date-year*10000)/100)
> gen day = date-year*10000-month*100
```

Data qualifiers are described in the website: <https://aqs.epa.gov/aqsweb/codes/data/QualifierCodes.html>

Using this description, you can check whether or not you need to drop some observations, after which you can drop the qualifiers so that the dataset is not as large. All the informational qualifiers start with ‘I’, whereas the rest of the qualifiers are either numbers or start with a different letter. Also, if the qualifier information is missing, then this means that the data had no comments. Then, we can run the following commands to keep the observations that had either no comments or that had only informational comments.

```
> keep if missing(qualifier1) | (substr(qualifier1,1,1)=="I")
> keep if missing(qualifier2) | (substr(qualifier2,1,1)=="I")
> keep if missing(qualifier3) | (substr(qualifier3,1,1)=="I")
> keep if missing(qualifier4) | (substr(qualifier4,1,1)=="I")
> keep if missing(qualifier5) | (substr(qualifier5,1,1)=="I")
> keep if missing(qualifier6) | (substr(qualifier6,1,1)=="I")
> keep if missing(qualifier7) | (substr(qualifier7,1,1)=="I")
> keep if missing(qualifier8) | (substr(qualifier8,1,1)=="I")
> keep if missing(qualifier9) | (substr(qualifier9,1,1)=="I")
```

```
> keep if missing(qualifier10) | (substr(qualifier10,1,1)=="I")
```

If you get a 'type mismatch' error, then this could mean that there are no observations for one or more qualifier columns. Since they are no longer needed, we can drop these qualifier variables.

```
> drop qualifier1 qualifier2 qualifier3 qualifier4 qualifier5 qualifier6 qualifier7 qualifier8 qualifier9
qualifier10
```

One of the most important variables is "unit." It may be the case that the concentrations are stored in different units, thus it is very important that we convert units to the most appropriate unit, which in the case of PM<sub>10</sub> is µg/m<sup>3</sup>. We can run the following command to get to know if we are dealing with one unit or more than one:

```
> tab unit
```

Then, once we have identified the units we can begin converting the units. The unit codes can be found in the following website:

<https://aqs.epa.gov/aqsweb/codes/data/Parameters-ALL.html>

The most common units to handle in PM<sub>10</sub> datasets are the following:

| Table 1    |                            |
|------------|----------------------------|
| Unit codes |                            |
| Unit code  | Concentration unit         |
| 1          | Micrograms per cubic meter |
| 5          | Miligrams per cubic meter  |

Source: <https://aqs.epa.gov/aqsweb/codes/data/Parameters-ALL.html>

Therefore, if we want to transform all the concentrations to µg/m<sup>3</sup>, we need to run the following commands:

```
> drop if missing(PM10)
> replace PM10 = PM10*1000 if unit==5
> drop unit
> label variable PM10 "PM10 in ug/m3"
```

Now, we can analyze the data by sampling frequencies and sample durations. These two variables and their codes are described in the following websites:

<https://aqs.epa.gov/aqsweb/codes/data/CollectionFrequencies.html>

<https://aqs.epa.gov/aqsweb/codes/data/SampleDurationCodes.html>

In some instances, some observations may have missing values for sampling frequency, if that is the case, we need to replace them with "0" so that we keep track of them when running the tab command.

```
> replace samplingfrequency=0 if missing(samplingfrequency)
```

Then, we can run 'tab' commands in order to identify with how many sampling frequencies and sample durations we are dealing. After doing this, we can start creating averages and dropping values if they don't comply with the 75% rule (observations with less than 75% of observations during the averaging period are assumed to yield averages not representatives of that period).

```
> tab samplingfrequency sampleduration
> collapse (mean) PM10 , by(statecode countycode siteid sampleduration samplingfrequency year month day hour)
> collapse (mean) PM10 (count) hour , by(statecode countycode siteid sampleduration samplingfrequency year month
day)
> tab hour
```

```
> drop if(hour<18&sampleduration==1)
```

By running this command we have now daily averages of the hourly data and the data that initially had daily averages hasn't been modified in any way, so far. Now, we need to create new variables that store information regarding the number of samples recorded by each monitoring station per sampling frequency and sample duration. To do this, I created Boolean variables (True=1, False=0) related to the sampling frequencies and sample durations:

```
> gen byte boolean_sf0 = samplingfrequency==0
> gen byte boolean_sf1 = samplingfrequency==1
> gen byte boolean_sf2 = samplingfrequency==2
> gen byte boolean_sf3 = samplingfrequency==3
> gen byte boolean_sf6 = samplingfrequency==6
> gen byte boolean_sf7 = samplingfrequency==7
> gen byte boolean_sf9 = samplingfrequency==9
> gen byte boolean_sf11 = samplingfrequency==11
> gen byte boolean_sd1 = sampleduration==1
> gen byte boolean_sd7 = sampleduration==7
```

Now, we can start creating daily and monthly averages combining all the samples, regardless of the sampling frequency or sample duration, but still keeping this information since it will be useful when dropping observations depending on the number of samples recorded.

```
> collapse (mean) PM10 (sum) boolean_sf0 boolean_sf1 boolean_sf2 boolean_sf3 boolean_sf6 boolean_sf7 boolean_sf9
boolean_sf11 boolean_sd1 boolean_sd7, by(statecode countycode siteid year month day)
> collapse (mean) PM10 (sum) boolean_sf0 boolean_sf1 boolean_sf2 boolean_sf3 boolean_sf6 boolean_sf7 boolean_sf9
boolean_sf11 boolean_sd1 boolean_sd7, by(statecode countycode siteid year month)
```

Now we can start dropping observations that do not comply with the 75% rule. However, in this case it will be easier to keep the observations that comply with the 75% rule, since there are some means that are comprised of various samples at different frequencies and/or different sample durations.

```
> keep if(((boolean_sd1>=21 | boolean_sf1>=21 | boolean_sf2>=10 | boolean_sf3>=7 | boolean_sf6>=3 |
boolean_sf7>=2)&month==2) | ((boolean_sd1>=22 | boolean_sf1>=22 | boolean_sf2>=11 | boolean_sf3>=7 |
boolean_sf6>=3 | boolean_sf7>=2)&(month==4 | month==6 | month==9 | month==11)) | ((boolean_sd1>=23 |
boolean_sf1>=23 | boolean_sf2>=12 | boolean_sf3>=8 | boolean_sf6>=4 | boolean_sf7>=2)&(month==1 | month==3 |
month==5 | month==7 | month==8 | month==10 | month==12)))
```

By running the previous command we are keeping the monthly observations that comply with the 75% rule for, at least, one sampling frequency. Then we can collapse the data in order to obtain annual means.

```
> collapse (mean) PM10 (count) month, by(statecode countycode siteid year)
> drop if month<9
> collapse (mean) PM10, by(statecode countycode siteid)
> label variable PM10 "Mean annual PM10 concentration in ug/m3"
> gen FIPS = statecode*1000+countycode
> order FIPS, first
> drop statecode countycode
> save "E:\Research\EPA-AirQuality\PM10\PM10_2010.dta", replace
```

Note:

Bravo et al. 2012. "Comparison in exposure estimation methods for air pollutants: Ambient monitoring data and regional air quality simulation." *Environmental Research* 116 (2012) 1-10.

This article excludes PM<sub>2.5</sub> data using the criteria of one-in-three-day sampling frequency (121 days a year). If they had less than 76% complete data (i.e., fewer than 91 observations), the record was excluded. Monitors were also excluded if they had fewer than 11 (of 13) 28-day periods with at least one observation per week for three or more weeks in the period. Inclusion criteria for O<sub>3</sub> were based on a daily measurement frequency during April-September. They included only monitors with 8-hour maximum reported for a minimum of 75% of

days in April through September. They also required that each monitor should have at least 50% of day concentrations for each month for 5 or more of the 6 months.

### Monitoring Site location

However, since there may be several stations within one county and one may be more representative of the population by being closer to denser areas, for example, it is more appropriate to save the means by FIPS and siteid and then plot them spatially. Once we obtain the spatial distribution of concentrations, one is able to obtain the concentration at the population-weighted centroid, by performing a spatial analysis to be later determined.

To perform this, we need location information of the monitoring sites. This data can be downloaded from: <http://www.epa.gov/ttn/airs/airsaqs/detaildata/downloadaqsdta.htm>

The link in this website is called "site and monitor records." This data is in txt format. To import it to Stata one can run File\Import\Text created by a spreadsheet. Then, select as the Delimiter as User-specified delimiter and use the character: |.

```
> insheet using "E:\Research\EPA-AirQuality\station_location\sites.txt", delimiter("|") clear
```

Once the data is imported, you will have 45 variables. However, this file contains information about the location and the parameters collected by each station so only some variables and only some observations are of interest to the current analysis. So, in order to keep the variables and observations of interest one can run the following commands:

```
> keep if (v1=="AA")
> keep v3 v4 v5 v6 v7
> rename v3 State
> rename v4 County
> rename v5 siteid
> rename v6 latitude
> rename v7 longitude
> destring longitude , replace
> drop if missing(longitude)|missing(latitude)|longitude==0|latitude==0
> label variable latitude "Latitude in degrees"
> label variable longitude "Longitude in degrees"
> gen FIPS = State*1000+County
> order FIPS , first
> drop State County
> save "E:\Research\EPA-AirQuality\station_location\sites.dta"
```

Then we can merge these two datasets.

```
> use "E:\Research\EPA-AirQuality\PM10\PM10_2010.dta", clear
> merge 1:1 FIPS siteid using "E:\Research\EPA-AirQuality\station_location\sites.dta"
> drop if _merge==2
```

You will notice that there are site monitoring stations that have no location information (\_merge==1). Since I don't want to lose this data, I am assuming that the mean concentration within one county is equal to the mean concentration across the stations within this county. In order to do this, I ran the following commands:

```
> replace latitude = 0 if missing(latitude)
> replace longitude = 0 if missing(longitude)
> collapse (mean) PM10 , by(FIPS latitude longitude)
```

Now we have some 'artificial' stations with no location information. Since we will need to plot later, we need to assign them artificial locations as well. Their assigned locations will be the corresponding population-weighted county centroid.

```
> rename latitude lat_station
> rename longitude long_station
> merge m:1 FIPS using "E:\Research\Population\PopCentroidCounty\County_Popcentroid.dta"
> keep if _merge==3
> replace lat_station = latitude if lat_station==0
> replace long_station = longitude if long_station==0
> keep FIPS lat_station long_station PM10
> rename lat_station latitude
> rename long_station longitude
> label variable PM10 "Mean annual PM10 concentration in ug/m3"
> sort FIPS
> save "E:\Research\EPA-AirQuality\PM10\PM10_2010_geo.dta"
> export excel using "E:\Research\EPA-AirQuality\PM10\PM10_2010_geo.xls", firstrow(variables)
```

Even though it is expected to see air quality concentrations to vary spatially, using long lat as explanatory variables in the parametric model may yield results relatively far from reality since the 'spatial' variation of concentrations accounted for by the parametric model would be influenced by the selection bias of having discrete monitoring locations, most of them located on urban areas. Therefore, in general, areas in which sampling was performed on urban areas would yield higher concentrations on non-urban areas simply because they have similar geographic locations. Because of this uncertainty, I decided to use ordinary kriging as the spatial interpolation method.

Concentrations may be compared with the National Ambient Air Quality Standards

<http://www.epa.gov/air/criteria.html#2>.

Air Quality Index may be used as well.

Definition: <http://airnow.gov/index.cfm?action=aqibasics.aqi>

Source (1970 - 2012): <http://www.epa.gov/ttn/airs/airsaqs/detaildata/AQIindex.htm>

For future references and comparisons, air quality concentrations can be downloaded directly from the following webpage:

For 2000: <http://www.epa.gov/airtrends/reports.html>

For 2010: <http://www.epa.gov/airtrends/factbook.html>

By downloading the air quality statistics by county for both years, we can make comparisons directly with the NAAQS and we make sure that the values used in the analyses correspond to the EPA values.

## 36. Ozone

See PM2.5 (Section 34).

### 2008 Downscaling model

```
> insheet using "E:\Research\EPA-AirQuality\O3\data\Downscaling\Predictions08ozone-Census2010edited.csv", clear
> rename v1 date, replace
> rename v2 FIPS, replace
> rename v3 latitude, replace
> rename v4 longitude, replace
> rename v5 O3, replace
> rename v6 std_error, replace
> collapse O3 , by(FIPS)
> label variable O3 "Mean annual O3 concentration in ppb"
> label variable FIPS "Census tract FIPS code"
> merge 1:1 FIPS using "E:\Research\Population\pop_censustract2010.dta"
```

```
> drop if State==2|State==15|State>56
> sum Population if(_merge==2)
> keep if _merge==3
> keep O3 State County Population
> gen FIPS = State*1000+County
> drop State County
> gen popO3 = Population*O3
> collapse (sum) popO3 Population , by(FIPS)
> gen O3 = popO3/Population
> keep FIPS O3
> label variable O3 "Mean annual O3 concentration in ppb"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> save "E:\Research\EPA-AirQuality\O3\O3_2008.dta"
```

## 2001 Downscaling model

Pending

## 2000 Detailed Air Quality System Data

```
> insheet using "E:\Research\EPA-AirQuality\O3\data\AQ5\RD_501_44201_2000-0.txt", delimiter("|") clear
> tab rd
> keep if rd == "RD"
> tab actioncode
> tab parameter
> drop rd actioncode parameter
> rename samplevalue O3
> split starttime, parse(:)
> drop starttime
> rename starttime1 hour
> rename starttime2 minute
> destring hour minute , replace
> drop if(statecode=="CC")
> destring , replace
> drop if(statecode==2|statecode==15|statecode>56)
> destring , replace
> gen year = floor(date/10000)
> gen month = floor((date-year*10000)/100)
> gen day = date-year*10000-month*100
> keep if missing(qualifier1)|(substr(qualifier1,1,1)=="I")
> keep if missing(qualifier2)|(substr(qualifier2,1,1)=="I")
> keep if missing(qualifier3)|(substr(qualifier3,1,1)=="I")
> keep if missing(qualifier4)|(substr(qualifier4,1,1)=="I")
> keep if missing(qualifier5)|(substr(qualifier5,1,1)=="I")
> keep if missing(qualifier6)|(substr(qualifier6,1,1)=="I")
> keep if missing(qualifier7)|(substr(qualifier7,1,1)=="I")
> keep if missing(qualifier8)|(substr(qualifier8,1,1)=="I")
> keep if missing(qualifier9)|(substr(qualifier9,1,1)=="I")
> keep if missing(qualifier10)|(substr(qualifier10,1,1)=="I")
> drop if qualifier1>=0&qualifier1!=.
> drop if qualifier2>=0&qualifier2!=.
> drop if qualifier3>=0&qualifier3!=.
> drop if qualifier4>=0&qualifier4!=.
> drop if qualifier5>=0&qualifier5!=.
> drop if qualifier6>=0&qualifier6!=.
> drop if qualifier7>=0&qualifier7!=.
> drop if qualifier8>=0&qualifier8!=.
> drop if qualifier9>=0&qualifier9!=.
> drop if qualifier10>=0&qualifier10!=.
> drop qualifier1 qualifier2 qualifier3 qualifier4 qualifier5 qualifier6 qualifier7 qualifier8 qualifier9
qualifier10
> tab unit
> drop if missing(O3)
> drop if O3<0
> replace O3 = O3*1000 if unit==7
> replace O3 = O3*10 if unit==40
> replace O3 = O3*100 if unit==87
> label variable O3 "O3 in ppb"
> drop unit
> replace samplingfrequency=0 if missing(samplingfrequency)
> tab samplingfrequency sampleduration
```

```
> collapse (mean) O3 , by(statecode countycode siteid year month day hour)
> gen O3_0 = O3
> gen O3_1 = O3
> gen O3_2 = O3
> gen O3_3 = O3
> gen O3_4 = O3
> gen O3_5 = O3
> gen O3_6 = O3
> gen O3_7 = O3
> gen O3_8 = O3
> gen O3_9 = O3
> gen O3_10 = O3
> gen O3_11 = O3
> gen O3_12 = O3
> gen O3_13 = O3
> gen O3_14 = O3
> gen O3_15 = O3
> gen O3_16 = O3
> replace O3_0 = . if hour>=8
> replace O3_1 = . if hour==0|hour>=9
> replace O3_2 = . if hour<=1|hour>=10
> replace O3_3 = . if hour<=2|hour>=11
> replace O3_4 = . if hour<=3|hour>=12
> replace O3_5 = . if hour<=4|hour>=13
> replace O3_6 = . if hour<=5|hour>=14
> replace O3_7 = . if hour<=6|hour>=15
> replace O3_8 = . if hour<=7|hour>=16
> replace O3_9 = . if hour<=8|hour>=17
> replace O3_10 = . if hour<=9|hour>=18
> replace O3_11 = . if hour<=10|hour>=19
> replace O3_12 = . if hour<=11|hour>=20
> replace O3_13 = . if hour<=12|hour>=21
> replace O3_14 = . if hour<=13|hour>=22
> replace O3_15 = . if hour<=14|hour>=23
> replace O3_16 = . if hour<=15|hour==24
> gen O3_0c = 1 if O3_0!=.
> gen O3_1c = 1 if O3_1!=.
> gen O3_2c = 1 if O3_2!=.
> gen O3_3c = 1 if O3_3!=.
> gen O3_4c = 1 if O3_4!=.
> gen O3_5c = 1 if O3_5!=.
> gen O3_6c = 1 if O3_6!=.
> gen O3_7c = 1 if O3_7!=.
> gen O3_8c = 1 if O3_8!=.
> gen O3_9c = 1 if O3_9!=.
> gen O3_10c = 1 if O3_10!=.
> gen O3_11c = 1 if O3_11!=.
> gen O3_12c = 1 if O3_12!=.
> gen O3_13c = 1 if O3_13!=.
> gen O3_14c = 1 if O3_14!=.
> gen O3_15c = 1 if O3_15!=.
> gen O3_16c = 1 if O3_16!=.
> collapse (mean) O3_0 O3_1 O3_2 O3_3 O3_4 O3_5 O3_6 O3_7 O3_8 O3_9 O3_10 O3_11 O3_12 O3_13 O3_14 O3_15 O3_16
(sum) O3_0c O3_1c O3_2c O3_3c O3_4c O3_5c O3_6c O3_7c O3_8c O3_9c O3_10c O3_11c O3_12c O3_13c O3_14c O3_15c
O3_16c , by(statecode countycode siteid year month day)
> replace O3_0 = 0 if
```

## 37. Sulphur dioxide

There is air quality data from 1993 to 2012, available for download from:

<http://www.epa.gov/ttn/airs/airsaqs/detaildata/downloadaqsdata.htm>. Data was downloaded for years 2000 and 2010 for the following parameters:

- PM2.5,
- PM10,



- Nitrogen dioxide,
- Sulfur dioxide,
- Carbon Monoxide, and,
- Ozone.

The following steps describe how to work with the CO, SO2 and NO2 data.

Data is available in concentrations reported at an hourly frequency, in txt format. In order to import the data into Stata, it was necessary to import it using the prompt “File/Import/Text data created by the spreadsheet.” However, before importing, one needs to make sure that the first line in the .txt file names each variable. If this is not the case, then editing the .txt file and naming each column is crucial. Open the .txt file and identify the character used to separate columns. In this case, the character was ‘|’; it is always crucial to identify this character.

After selecting the option “File/Import/Text data created by the spreadsheet”, a window pops up. In this window, we have to load the .txt file, and in the delimiter section, we have to select “User-specified delimiter” and type | in the box just below it.

```
> insheet using "E:\Research\EPA-AirQuality\SO2\data\AQS\RD_501_42401_1H_2010-0.txt", delimiter("|") clear
```

After importing the file, we can get rid of the irrelevant variables. To erase these variables, we can run the ‘drop’ command (e.g., drop rd). I dropped the variables that had the same value for each observation, such as ‘rd’, and variables that had no information, such as qualifier4. To have a quick idea of the observations stored in each variable, one can run the commands ‘sum \*variable name\*’ or ‘tab \*variable name\*’. After viewing the variables and the data they stored I dropped some variables.

```
> tab rd
> keep if rd=="RD"
> tab actioncode
> tab parameter
> drop rd actioncode parameter
> rename samplevalue SO2
```

The start time variable may have been stored as a string variable; if this is the case, then we can run the following commands to convert it into a numeric variable.

```
> split starttime, parse(:)
> drop starttime
> rename starttime1 hour
> rename starttime2 minute
> destring hour , replace
> destring minute , replace
```

In some instances, the state code will be a string variable because it has some observations recorded as “CC”. When this is the case, we need to drop the observations in which the state code is equal to CC. Run the following command:

```
> drop if(statecode="CC")
```

Then, destring the statecode variable by running the following command:

```
> destring statecode, replace
```

Since we are only analyzing the counties for the conterminous US, we need to drop all the observations whose statecode correspond to Alaska, Hawaii, Puerto Rico, Virgin Islands, and another area with FIPS code equal to 80; run the following command:

```
> drop if (statecode==2|statecode==15|statecode==72|statecode==78|statecode==80)
```

### Obtaining means and running the quality control process

To create the means first we need to create year, month and day variables. To do this we can run the following commands, assuming that the variable “date” stores the year, month and day information with the first four characters for the year, the next two for month and the next two for day:

```
> gen year = floor(date/10000)
> gen month = floor((date-year*10000)/100)
> gen day = date-year*10000-month*100
```

Data qualifiers are described in the website: <https://aqs.epa.gov/aqsweb/codes/data/QualifierCodes.html>

Using this description, you can check whether or not you need to drop some observations, after which you can drop the qualifiers so that the dataset is not as large. All the informational qualifiers start with ‘I’, whereas the rest of the qualifiers are either numbers or start with a different letter. Also, if the qualifier information is missing, then this means that the data had no comments. Then, we can run the following commands to keep the observations that had either no comments or that had only informational comments.

```
> keep if missing qualifier1 | (substr qualifier1,1,1)=="I"
> keep if missing qualifier2 | (substr qualifier2,1,1)=="I"
> keep if missing qualifier3 | (substr qualifier3,1,1)=="I"
> keep if missing qualifier4 | (substr qualifier4,1,1)=="I"
> keep if missing qualifier5 | (substr qualifier5,1,1)=="I"
> keep if missing qualifier6 | (substr qualifier6,1,1)=="I"
> keep if missing qualifier7 | (substr qualifier7,1,1)=="I"
> keep if missing qualifier8 | (substr qualifier8,1,1)=="I"
> keep if missing qualifier9 | (substr qualifier9,1,1)=="I"
> keep if missing qualifier10 | (substr qualifier10,1,1)=="I"
```

If you get a ‘type mismatch’ error, then this could mean that there are no observations for one or more qualifier columns. Since they are no longer needed, we can drop these qualifier variables.

```
> drop qualifier1 qualifier2 qualifier3 qualifier4 qualifier5 qualifier6 qualifier7 qualifier8 qualifier9
qualifier10
```

One of the most important variables is “unit.” It may be the case that the concentrations are stored in different units, thus it is very important that we convert units to the most appropriate unit, which could be ppm, ppb or  $\mu\text{g}/\text{m}^3$ , depending on the parameter being analyzed; in the case of SO<sub>2</sub>, the most appropriate unit is ppb. We can run the following command to get to know if we are dealing with one unit or more than one:

```
> tab unit
```

Then, once we have identified the units we can begin converting the units to, for instance, ppb. The unit codes can be found in the following website:

<https://aqs.epa.gov/aqsweb/codes/data/Parameters-ALL.html>

The most common units to handle in these datasets are the following:

**Table 2**  
**Unit codes**

| Unit code | Concentration unit |
|-----------|--------------------|
|-----------|--------------------|

|    |                       |
|----|-----------------------|
| 7  | Parts per million     |
| 8  | Parts per billion     |
| 40 | Parts per 100 million |
| 87 | Parts per 10 million  |

Source: <https://aqs.epa.gov/aqsweb/codes/data/Parameters-ALL.html>

Therefore, if we want to transform all the concentrations to ppb, we need to run the following commands:

```
> drop if missing(SO2)
> replace SO2 = SO2*1000 if unit==7
> replace SO2 = SO2*10 if unit==40
> replace SO2 = SO2*100 if unit==87
> label variable SO2 "SO2 in ppb"
> drop unit
```

Now, we can analyze the data by sampling frequencies and sample durations. These two variables and their codes are described in the following websites:

<https://aqs.epa.gov/aqsweb/codes/data/CollectionFrequencies.html>

<https://aqs.epa.gov/aqsweb/codes/data/SampleDurationCodes.html>

In the case of the 2010 SO2 dataset, it only has 1-hour duration samples (sampleduration=1, null samplingfrequency). Therefore, there is no need to differentiate among observations by sampling frequency or sample duration since we know that all the records are observations recorded at an hourly frequency. Assuming that we need at least 75% of the data in a given period to consider this data as representative of the averaging period, we would need to drop the observations with less than 18 hours a day, 23 days a month and 9 months a year. To do this, we can run the following commands:

```
> collapse (mean) SO2 , by(statecode countycode siteid year month day hour)
> collapse (mean) SO2 (count) hour , by(statecode countycode siteid year month day)
> tab hour
> drop if hour<18
> collapse (mean) SO2 (count) day , by(statecode countycode siteid year month)
> tab day
> drop if (month==2&day<21)
> drop if ( (month==1|month==3|month==5|month==7|month==8|month==10|month==12) &day<23)
> drop if ( (month==4|month==6|month==9|month==11) &day<22)
> collapse (mean) SO2 (count) month , by(statecode countycode siteid year)
> tab month
> drop if month<9
> collapse (mean) SO2 , by(statecode countycode siteid)
> label variable SO2 "Mean annual SO2 concentration in ppb"
> gen FIPS = statecode*1000+countycode
> order FIPS , first
> drop statecode countycode
> save "E:\Research\EPA-AirQuality\SO2\SO2_2010.dta", replace
```

## Monitoring Site location

However, since there may be several stations within one county and one station may be more representative of the population by being closer to denser areas, it is more appropriate to save the means by FIPS and siteid and then plot them spatially. Once we obtain the spatial distribution of concentrations, one is able to obtain the concentration at the county population-weighted centroid, by performing a spatial analysis.

To perform this, we need location information of the monitoring sites. This data can be downloaded from:

<http://www.epa.gov/ttn/airs/airsaqs/detaildata/downloadaqsdta.htm>

We already imported this file in PM10-Section 35, thus this procedure will not be addressed in this section.

We need to merge the monitoring site datasets with the SO2 dataset.

```
> use "E:\Research\EPA-AirQuality\SO2\SO2_2010.dta", clear
> merge 1:1 FIPS siteid using "E:\Research\EPA-AirQuality\station_location\sites.dta"
> drop if _merge==2
```

You will notice that there are site monitoring stations that have no location information (`_merge==1`). Since I don't want to lose this data, I am assuming that the mean concentration within one county is equal to the mean concentration across the stations within this county. In order to do this, I ran the following commands:

```
> replace latitude = 0 if missing(latitude)
> replace longitude = 0 if missing(longitude)
> collapse (mean) SO2 , by(FIPS latitude longitude)
```

Now we have some 'artificial' stations with no location information. Since we will need to plot later, we need to assign them artificial locations as well. Their assigned locations will be the corresponding population-weighted county centroid.

```
> rename latitude lat_station
> rename longitude long_station
> merge m:1 FIPS using "E:\Research\Population\PopCentroidCounty\County_Popcentroid.dta"
> keep if _merge==3
> replace lat_station = latitude if lat_station==0
> replace long_station = longitude if long_station==0
> keep FIPS lat_station long_station SO2
> rename lat_station latitude
> rename long_station longitude
> label variable SO2 "Mean annual SO2 concentration in ppb "
> sort FIPS
> save "E:\Research\EPA-AirQuality\SO2\SO2_2010_geo.dta"
> export excel using "E:\Research\EPA-AirQuality\SO2\SO2_2010_geo.xls", firstrow(variables)
```

Finally, see final section of PM10 (Section 35) to obtain the mean annual SO2 concentration at the county population-weighted centroids.

## 38. Carbon Monoxide

The process is very similar to the one followed for Sulphur dioxide (Section 37). However, since CO's preferred unit is ppm, we'll need to transform units to ppm, instead of to ppb. The commands are the following:

First, we need to import the txt file into Stata and format it.

```
> insheet using "E:\Research\EPA-AirQuality\CO\data\AQS\RD_501_42101_2010-0.txt", delimiter("|") clear
> tab rd
> keep if rd=="RD"
> tab actioncode
> tab parameter
> drop rd actioncode parameter
> rename samplevalue CO
> split starttime, parse(:)
> drop starttime
> rename starttime1 hour
> rename starttime2 minute
> destring hour , replace
> destring minute , replace
> drop if(statecode=="CC")
> destring statecode, replace
> drop if(statecode==2|statecode==15|statecode==72|statecode==78|statecode==80)
```

## Obtaining means and running the quality control process

```
> gen year = floor(date/10000)
> gen month = floor((date-year*10000)/100)
> gen day = date-year*10000-month*100
> keep if missing(qualifier1) | (substr(qualifier1,1,1)=="I")
> keep if missing(qualifier2) | (substr(qualifier2,1,1)=="I")
> keep if missing(qualifier3) | (substr(qualifier3,1,1)=="I")
> keep if missing(qualifier4) | (substr(qualifier4,1,1)=="I")
> keep if missing(qualifier5) | (substr(qualifier5,1,1)=="I")
> keep if missing(qualifier6) | (substr(qualifier6,1,1)=="I")
> keep if missing(qualifier7) | (substr(qualifier7,1,1)=="I")
> keep if missing(qualifier8) | (substr(qualifier8,1,1)=="I")
> keep if missing(qualifier9) | (substr(qualifier9,1,1)=="I")
> keep if missing(qualifier10) | (substr(qualifier10,1,1)=="I")
> drop qualifier1 qualifier2 qualifier3 qualifier4 qualifier5 qualifier6 qualifier7 qualifier8 qualifier9
qualifier10
> tab unit
```

Then, once we have identified the units we can begin converting the units to, for instance, ppm. The most common units to handle in these datasets are the following:

**Table 2**  
**Unit codes**

| Unit code | Concentration unit    |
|-----------|-----------------------|
| 7         | Parts per million     |
| 8         | Parts per billion     |
| 40        | Parts per 100 million |
| 87        | Parts per 10 million  |

Source: <https://aqs.epa.gov/aqsweb/codes/data/Parameters-ALL.html>

We need to run the following commands to transform all concentrations to ppm:

```
> drop if missing(CO)
> replace CO = CO/1000 if unit==8
> replace CO = CO/100 if unit==40
> replace CO = CO/10 if unit==87
> label variable CO "CO in ppm"
> drop unit
```

In the case of the 2010 CO dataset, it only has 1-hour duration samples (sampleduration=1, null samplingfrequency). Therefore, there is no need to differentiate among observations by sampling frequency or sample duration since we know that all the records are observations recorded at an hourly frequency. Assuming that we need at least 75% of the data in a given period to consider this data as representative of the averaging period, we would need to drop the observations with less than 18 hours a day, 23 days a month and 9 months a year. To do this, we can run the following commands:

```
> collapse (mean) CO , by(statecode countycode siteid year month day hour)
> collapse (mean) CO (count) hour , by(statecode countycode siteid year month day)
> tab hour
> drop if hour<18
> collapse (mean) CO (count) day , by(statecode countycode siteid year month)
> tab day
> drop if (month==2&day<21)
> drop if ((month==1|month==3|month==5|month==7|month==8|month==10|month==12)&day<23)
> drop if ((month==4|month==6|month==9|month==11)&day<22)
> collapse (mean) CO (count) month , by(statecode countycode siteid year)
> tab month
> drop if month<9
> collapse (mean) CO , by(statecode countycode siteid)
> label variable CO "Mean annual CO concentration in ppm"
> gen FIPS = statecode*1000+countycode
```

```
> order FIPS , first
> drop statecode countycode
> save "E:\Research\EPA-AirQuality\CO\CO_2010.dta", replace
```

## Monitoring Site location

We already imported this file in PM10-Section 35, thus this procedure will not be addressed in this section. We need to merge the monitoring site datasets with the CO dataset.

```
> use "E:\Research\EPA-AirQuality\CO\CO_2010.dta", clear
> merge 1:1 FIPS siteid using "E:\Research\EPA-AirQuality\station_location\sites.dta"
> drop if _merge==2
> replace latitude = 0 if missing(latitude)
> replace longitude = 0 if missing(longitude)
> collapse (mean) CO , by(FIPS latitude longitude)
> rename latitude lat_station
> rename longitude long_station
> merge m:1 FIPS using "E:\Research\Population\PopCentroidCounty\County_Popcentroid.dta"
> keep if _merge==3
> replace lat_station = latitude if lat_station==0
> replace long_station = longitude if long_station==0
> keep FIPS lat_station long_station CO
> rename lat_station latitude
> rename long_station longitude
> label variable CO "Mean annual CO concentration in ppm"
> sort FIPS
> save "E:\Research\EPA-AirQuality\CO\CO_2010_geo.dta", replace
> export excel using "E:\Research\EPA-AirQuality\CO\CO_2010_geo.xls", firstrow(variables) replace
```

Finally, see final section of PM10 (Section 35) to obtain the mean annual CO concentration at the county population-weighted centroids.

## 39. Nitrogen dioxide

The process is very similar to the one followed for Sulphur dioxide (Section 37). The commands are the following:

First, we need to import the txt file into Stata and format it.

```
> insheet using "E:\Research\EPA-AirQuality\NO2\data\AQ5\RD_501_42602_2010-0.txt", delimiter("|") clear
> tab rd
> keep if rd=="RD"
> tab actioncode
> tab parameter
> drop rd actioncode parameter
> rename samplevalue NO2
> split starttime, parse(:)
> drop starttime
> rename starttime1 hour
> rename starttime2 minute
> destring hour , replace
> destring minute , replace
> drop if(statecode=="CC")
> destring statecode, replace
> drop if(statecode==2|statecode==15|statecode==72|statecode==78|statecode==80)
```

## Obtaining means and running the quality control process

```
> gen year = floor(date/10000)
> gen month = floor((date-year*10000)/100)
> gen day = date-year*10000-month*100
> keep if missing(qualifier1)|(substr(qualifier1,1,1)=="I")
> keep if missing(qualifier2)|(substr(qualifier2,1,1)=="I")
> keep if missing(qualifier3)|(substr(qualifier3,1,1)=="I")
> keep if missing(qualifier4)|(substr(qualifier4,1,1)=="I")
> keep if missing(qualifier5)|(substr(qualifier5,1,1)=="I")
```

```
> keep if missing(qualifier6) | (substr(qualifier6,1,1)=="I")
> keep if missing(qualifier7) | (substr(qualifier7,1,1)=="I")
> keep if missing(qualifier8) | (substr(qualifier8,1,1)=="I")
> keep if missing(qualifier9) | (substr(qualifier9,1,1)=="I")
> keep if missing(qualifier10) | (substr(qualifier10,1,1)=="I")
> drop qualifier1 qualifier2 qualifier3 qualifier4 qualifier5 qualifier6 qualifier7 qualifier8 qualifier9
qualifier10
> tab unit
```

Then, once we have identified the units we can begin converting the units to, for instance, ppb. The most common units to handle in these datasets are the following:

**Table 2**  
**Unit codes**

| Unit code | Concentration unit    |
|-----------|-----------------------|
| 7         | Parts per million     |
| 8         | Parts per billion     |
| 40        | Parts per 100 million |
| 87        | Parts per 10 million  |

Source: <https://aqs.epa.gov/aqsweb/codes/data/Parameters-ALL.html>

We need to run the following commands to transform all concentrations to ppb:

```
> drop if missing(NO2)
> replace NO2 = NO2*1000 if unit==7
> replace NO2 = NO2*10 if unit==40
> replace NO2 = NO2*100 if unit==87
> label variable NO2 "NO2 in ppb"
> drop unit
```

In the case of the 2010 NO2 dataset, it only has 1-hour duration samples (sampleduration=1, null samplingfrequency). Therefore, there is no need to differentiate among observations by sampling frequency or sample duration since we know that all the records are observations recorded at an hourly frequency. Assuming that we need at least 75% of the data in a given period to consider this data as representative of the averaging period, we would need to drop the observations with less than 18 hours a day, 23 days a month and 9 months a year. To do this, we can run the following commands:

```
> collapse (mean) NO2 , by(statecode countycode siteid year month day hour)
> collapse (mean) NO2 (count) hour , by(statecode countycode siteid year month day)
> tab hour
> drop if hour<18
> collapse (mean) NO2 (count) day , by(statecode countycode siteid year month)
> tab day
> drop if (month==2&day<21)
> drop if ((month==1|month==3|month==5|month==7|month==8|month==10|month==12)&day<23)
> drop if ((month==4|month==6|month==9|month==11)&day<22)
> collapse (mean) NO2 (count) month , by(statecode countycode siteid year)
> tab month
> drop if month<9
> collapse (mean) NO2 , by(statecode countycode siteid)
> label variable NO2 "Mean annual NO2 concentration in ppb"
> gen FIPS = statecode*1000+countycode
> order FIPS , first
> drop statecode countycode
> save "E:\Research\EPA-AirQuality\NO2\NO2_2010.dta", replace
```

## Monitoring Site location

We already imported this file in PM10-Section 35, thus this procedure will not be addressed in this section. We need to merge the monitoring site datasets with the NO2 dataset.

```
> use "E:\Research\EPA-AirQuality\NO2\NO2_2010.dta", clear
> merge 1:1 FIPS siteid using "E:\Research\EPA-AirQuality\station_location\sites.dta"
> drop if _merge==2
> replace latitude = 0 if missing(latitude)
> replace longitude = 0 if missing(longitude)
> collapse (mean) NO2 , by(FIPS latitude longitude)
> rename latitude lat_station
> rename longitude long_station
> merge m:1 FIPS using "E:\Research\Population\PopCentroidCounty\County_Popcentroid.dta"
> keep if _merge==3
> replace lat_station = latitude if lat_station==0
> replace long_station = longitude if long_station==0
> keep FIPS lat_station long_station NO2
> rename lat_station latitude
> rename long_station longitude
> label variable NO2 "Mean annual NO2 concentration in ppb"
> sort FIPS
> save "E:\Research\EPA-AirQuality\NO2\NO2_2010_geo.dta"
> export excel using "E:\Research\EPA-AirQuality\NO2\NO2_2010_geo.xls", firstrow(variables)
```

Finally, see final section of PM10 (Section 35) to obtain the mean annual NO2 concentration at the county population-weighted centroids.

## 40. Non-attainment areas

The National Transportation Atlas Database (NTAD) 2010 from the Research and Innovative Technology Administration – Bureau of Transportation Statistics contains information regarding the level of attainment of air quality standards throughout the whole U.S. See Section 58 – Number of airports for obtaining these data. You should have 7 shapefiles of non-attainment areas, one for each criteria pollutant, i.e. O<sub>3</sub>, CO, NO<sub>2</sub>, SO<sub>2</sub>, PM<sub>10</sub>, PM<sub>2.5</sub> and Lead. Since these areas are polygons, we need to average this information to the county level. One way of doing this is obtaining the share of the population per county that is located within a non-attainment area. First, we need to work on ArcGIS to obtain this information at the block group level. The following could be used for all of the non-attainment areas shapefiles.

- (1) Add the non-attainment area shapefile and the blockgroup population-weighted centroid shapefile.
- (2) Right-click on the blockgroup shapefile and select “Join and Relates \ Join”.
  - a. What do you want to do with this layer: Join data from another layer based on spatial location
  - b. Choose the layer to join to this layer, or load spatial data from disk: non-attainment shapefile
  - c. You are joining: Polygons to Points.
  - d. Each point will be given all the attributes of the polygon that: it falls inside.
- (3) Then we need to export the attribute table of the newly created shapefile.

Then we can process this table in Stata.

```
> insheet using "E:\Research\USDT-NTAD\NonAttainment\data\blockgroup_naaCO.txt"
> gen naaCO = 0
> replace naaCO = 1 if status == "Nonattainment"
> gen FIPS = statefp*1000 + countyfp
> gen indexCO = population*naaCO
> collapse (sum) population indexCO , by(FIPS)
> gen naaCO = indexCO/population
> drop population indexCO
> label variable naaCO "Share of the population in the county in a CO non-attainment area"
> save "E:\Research\USDT-NTAD\NonAttainment\naaCO_2010.dta"
```

I have decided not to include the maintenance areas as non-attainment because a maintenance area has achieved compliance with the air quality standard although it is under strict monitoring during the maintenance period to



verify that is in fact complying with it. Since there is evidence that the area is no longer in non-compliance, I decided to consider them as compliance areas.<sup>7</sup>

## 41. Wildfire area (Previously National Fire Plan treatment)

I have contacted the Danny Lee, from the USDA Forest Service ([dclee@fs.fed.us](mailto:dclee@fs.fed.us)). According to him, the Forest Service is currently working on data analysis so this data is not yet ready for public distribution. However, he has agreed to send to me the data they are working currently working with. We might be able to use this data to calculate the area affected by wildfire at the county level.

Websites visited:

<http://www.usfa.fema.gov/statistics/estimates/wildfire.shtml>

[http://www.nifc.gov/fireInfo/fireInfo\\_statistics.html](http://www.nifc.gov/fireInfo/fireInfo_statistics.html)

National fire plan treatment can be accessed from

<http://nationalatlas.gov/atlasftp.html?openChapters=chpgeol%2Cchpbio#chpbio> or

<http://nationalatlas.gov/mld/firplnp.html>. The latest data is 2005. Raw data is stored in C:\Data\FirePlan. It has not been converted yet.

**Not converted into STATA yet.**

## 42. Cancer Risk

Data is available for years 1996, 1999, 2002 and 2005. We will use data from year 2005 as representative of 2010, data from year 2002 as representative of 2000, and data from year 1996 as representative from 1990. However, EPA mentions in its website (<http://www.epa.gov/ttn/atw/natamain/index.html>) that "due to the extent of improvements in methodology, it is not meaningful to compare the assessments. This is because any change in emissions, ambient concentrations, or risks may be due to either improvement in methodology or to real changes in emissions or source characterization." Therefore, these datasets should be used carefully. I also suggest running the model with and without these variables (i.e. cancer risk, neurological risk, respiratory risk) in order to assess the impact of considering these variables in the model.

### 2005 Data

Data for year 2005 is described in the following website: <http://www.epa.gov/ttn/atw/nata2005/>. The 2005 results can be downloaded from the following website: <http://www.epa.gov/ttn/atw/nata2005/tables.html>. Given that there is no data for Broomfield county, Colorado (FIPS 08014), it is necessary to work with the data at the census tract level and then take the data to the county level. By doing this, we will be able to select the tracts that belong to Broomfield county.

Once data at the tract-level is downloaded, it should be unzipped and loaded into Microsoft Access. Then, we can export the dataset as an Excel file and then import it into Stata. I only imported 6 variables: state, county, FIPS, tract, population, and CancerRisk (which stands for the Total Cancer Risk). Then I ran the following commands to format this dataset:

```
> drop if state=="AK"|state=="HI"|state=="PR"|state=="VI"  
> drop if county=="Nationwide"|county=="Statewide"
```

<sup>7</sup> [http://www.ecy.wa.gov/programs/air/sips/designations/maintenance\\_areas.htm](http://www.ecy.wa.gov/programs/air/sips/designations/maintenance_areas.htm)

```
> destring FIPS tract , replace
> rename state state_original
> rename county county_original
> sort FIPS tract
```

Now, since some FIPS codes may have changes, it is important to make some changes before merging this dataset with the FIPS dataset. In this case, it is needed to replace FIPS code 51560 (Clifton Forge, Virginia) for FIPS 51005 (Alleghany, Virginia) and to reassign some census tracts to the Broomfield county, Colorado (FIPS 08014).

```
> replace FIPS = 51005 if FIPS==51560
```

Before reassigning census tracts to Broomfield county it is important to save the dataset we've working on and then download the census tracts relationship files from the census website: <http://www.census.gov/geo/maps-data/data/relationship.html>. I downloaded the census tract relationship files for the State of Colorado by selecting Colorado from the 2010 Census Tract Relationship files option in the following website: [http://www.census.gov/geo/maps-data/data/tract\\_rel\\_download.html](http://www.census.gov/geo/maps-data/data/tract_rel_download.html). Now that you've downloaded the .txt file, you can import it into Stata.

```
> insheet state00 county00 tract00 geoid00 pop00 hu00 part00 area00 arealand00 state10 county10 tract10 geoid10
pop10 hu10 part10 area10 arealand10 areapt arealandpt areapt00pt arealandpct00pt araepct10pt arealandpct10pt
pop10pt poppct00 poppct10 hu10pt hupct00 hupct10 using "E:\Research\Tract_relationship\2010\co08trf.txt", comma
clear
> sort county10 tract10 county00 tract00
> keep state00 county00 tract00 pop00 part00 state10 county10 tract10 pop10 part10 poppct00 poppct10
> browse
```

Now, you'll see, for instance, that the census tract 13110 of FIPS 08013 from year 2000 was used wholly to create census 30000 of Broomfield county; thus we can use this whole census tract and changed its FIPS code to 08014 in the previously saved dataset (the one with cancer risk). Now, take a look at the field 'poppct00', this field represents the percentage of the population from the census tract from 2000 that was used to create the new 2010 census tract. Therefore, this 'poppct00' field is very important, since by multiplying this field with the 'population' field from the CancerRisk2005.dta file we can obtain the population that was 'transferred' to Broomfield county. Now, run the following commands to format the dataset and keep the necessary observations:

```
> keep if county10 == 14
> gen FIPS = state00*1000+county00
> gen FIPS10 = state10*1000+county10
> rename tract00 tract
```

Let's save this file and then merge it with the previously saved dataset (CancerRisk2005.dta); we'll use fields 'FIPS' and 'tract' as key fields. First we'll calculate the cancer risk for FIPS 08014.

```
> use E:\Research\EPA-NATA\2005\data\CancerRisk2005.dta
> merge m:m FIPS tract using "E:\Research\Tract_relationship\2010\08014.dta"
> keep if FIPS10 == 8014
> replace population = population*poppct00/100
> replace FIPS = FIPS10
> gen popindex = population*CancerRisk
> collapse (sum) popindex population , by(FIPS)
> gen CancerRisk = popindex/population*1000000
> drop popindex population
> save "E:\Research\EPA-NATA\2005\data\CancerRisk2005_08014_collapse.dta"
```

Now that the dta file for Broomfield County has been saved, we can work on the other counties. However, we need to work with the 2010\08014.dta before working with the other counties. This is because we need to create a dta file that has information regarding how many people were excluded from Broomfield County, not how

many people were excluded from each census tract within Broomfield County. If we don't do this, we might have double-counting errors.

```
> use "E:\Research\Tract_relationship\2010\08014.dta", clear
> collapse (sum) poppct00 , by(FIPS00 county00 tract00)
> replace poppct00 = 100- poppct00
> save "E:\Research\Tract_relationship\2010\outof08014.dta"
```

So, now this file (2010\outof08014.dta) has the percentage of population that remained outside of Broomfield County. We need to merge this file with the CancerRisk file previously created to calculate the risk on the other counties.

```
> use "E:\Research\EPA-NATA\2005\data\CancerRisk2005.dta", clear
> rename FIPS FIPS00
> rename tract tract00
> merge m:m FIPS00 tract00 using "E:\Research\Tract_relationship\2010\outof08014.dta"
> replace poppct00 = 100 if missing(poppct00)
> replace population = population*poppct00/100
> gen popindex = population*CancerRisk
> rename FIPS00 FIPS
> collapse (sum) popindex population , by(FIPS)
> gen CancerRisk = popindex/population*1000000
> drop popindex population
> save "E:\Research\EPA-NATA\2005\data\CancerRisk2005_but08014_collapse.dta"
```

Now that both files are saved, we can append them and then merge them with the FIPS.dta file.

```
> use E:\Research\EPA-NATA\2005\data\CancerRisk2005_but08014_collapse.dta
> append using "E:\Research\EPA-NATA\2005\data\CancerRisk2005_08014_collapse.dta"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> sort FIPS
> drop statecode _merge
> label variable CancerRisk "People at possible risk out of 1 million"
> save "E:\Research\EPA-NATA\2005\CancerRisk2005_collapse.dta"
```

## 2002 Data

One should follow almost the same procedure for the 2002 dataset. 2002 data at the census tract level was downloaded from <http://www.epa.gov/ttn/atw/nata2002/tables.html>. The only change, which is not significant at all, is the fact that when loading the Excel file, one should drop the observations where county is equal to 'ALL' or 'State' instead of 'Nationwide' or 'Statewide' (drop if county=="ALL"|county=="State").

## 1996 Data

Now, working with the 1996 dataset is a little trickier. 1996 data at the census tract level is available in Excel format, so there is no need to open it in Microsoft Access and export it to Excel. This data was downloaded from <http://www.epa.gov/ttn/atw/nata/ted/exporisk.html#agga>.

I only imported 6 variables: state, county, tract, co\_tract, population, and CancerRisk (which stands for the Total Cancer Risk). Then I ran the following commands to format this dataset:

```
> drop if state=="Alaska"|state=="Hawaii"|state=="Puerto Rico"|state=="Virgin Islands"
> rename county fullcounty
> gen countylength = length(fullcounty)
> gen countytype = substr(fullcounty, countylength-5, countylength)
> tab countytype
> gen county = substr(fullcounty, 1, countylength-7) if countytype=="County"|countytype=="Parish"
> replace county = substr(fullcounty, 1, countylength-6) if countytype==" County"|countytype==" Paris"
> replace countytype = "County" if countytype==" County"
> replace countytype = "Parish" if countytype==" Paris"
> replace county = fullcounty if (countytype==" . Mary"|countytype=="George"|countytype=="a city"|countytype=="aptist"|countytype=="d city"|countytype=="e city"|countytype=="g city"|countytype=="h city"|countytype=="k city"|countytype=="l city"|countytype=="lumbia"|countytype=="m city"|countytype=="n
```

```

Anne"|countytype=="n City"|countytype=="n city"|countytype=="o city"|countytype=="r city"|countytype=="s
city"|countytype=="tional"|countytype=="ts cit"|countytype=="x city")
> replace countytype = "" if (countytype==" . Mary"|countytype=="George"|countytype=="a
city"|countytype=="apist"|countytype=="d city"|countytype=="e city"|countytype=="g city"|countytype=="h
city"|countytype=="k city"|countytype=="l city"|countytype=="lumbia"|countytype=="m city"|countytype=="n
Anne"|countytype=="n City"|countytype=="n city"|countytype=="o city"|countytype=="r city"|countytype=="s
city"|countytype=="tional"|countytype=="ts cit"|countytype=="x city")
> tab county if countytype=="ts cit"
> replace county = "Colonial Heights city" if county=="Colonial Heights cit"
> replace county = substr(fullcounty,1,countylength-5) if countytype=="e Coun"|countytype=="k
Coun"|countytype=="o Coun"|countytype=="s Coun"|countytype=="s Pari"
> replace countytype = "County" if countytype=="e Coun"|countytype=="k Coun"|countytype=="o Coun"|countytype=="s
Coun"
> replace countytype = "Parish" if countytype=="s Pari"
> replace county = substr(fullcounty,1,countylength-3) if countytype=="ods Co"
> replace countytype = "County" if countytype=="ods Co"
> replace county = substr(fullcounty,1,countylength-4) if countytype=="ge Par"
> replace countytype = "Parish" if countytype=="ge Par"
> tab countytype
> drop countylength
> order county countytype tract co_tract , after(state)
> order fullcounty , last
> sort state county tract
> replace county = proper(county)
> replace state = proper(state)
> replace county = "Miami-Dade" if county=="Dade"&state=="Florida"
> replace county = "Lasalle" if county=="La Salle"&state=="Illinois"
> replace county = "DeKalb" if county=="De Kalb"&state=="Indiana"
> replace county = "Laporte" if county=="La Porte"&state=="Indiana"
> replace county = "Prince George's" if county=="Prince George"&state=="Maryland"
> replace county = "Queen Anne's" if county=="Queen Anne"&state=="Maryland"
> replace county = "St. Mary's" if county=="St. Mary"&state=="Maryland"
> replace county = "De Baca" if county=="DeBaca"&state=="New Mexico"
> replace county = "Doña Ana" if county=="Dona Ana"&state=="New Mexico"
> replace county = "McKean" if county=="Mc Kean"&state=="Pennsylvania"
> save "E:\Research\EPA-NATA\1996\data\CancerRisk1996.dta"

```

Now, since some FIPS codes may have changes, it is important to make some changes before merging this dataset with the FIPS dataset. In this case, it is needed to replace FIPS code 51560 (Clifton Forge, Virginia) for FIPS 51005 (Alleghany, Virginia) and to reassign some census tracts to the Broomfield county, Colorado (FIPS 08014).

```

> merge m:m state county using "E:\Research\FIPS\FIPS.dta"
> replace FIPS = 51005 if county=="Clifton Forge City"&state=="Virginia"
> save "E:\Research\EPA-NATA\1996\data\CancerRisk1996_mergeFIPS.dta"

```

Before reassigning census tracts to Broomfield County it is important to save the dataset we've working on and then create a new relationship file from 1990 census tracts to 2010 census tracts. We can use the 08014.dta file created for the 2005 dataset, however, we also need the census tracts relationship files for 1990 to 2000. We can download that data from [http://www.census.gov/geo/www/relate/rel\\_tract.html](http://www.census.gov/geo/www/relate/rel_tract.html) (<http://www2.census.gov/geo/relfiles/tract/co/co08pop.txt>).

Once the txt file is downloaded we need to import it into Stata. In order to do this I used a dictionary file with the following syntax:

```

infile dictionary {
_column(1) str2 state90 %2s "1990 state FIPS code"
_column(3) str3 county90 %3s "1990 county FIPS code"
_column(6) str4 tractbas90 %4s "1990 census tract base"
_column(10) str2 tractsuf90 %2s "1990 census tract suffix"
_column(12) str1 flag90 %1s "1990 census tract part flag"
_column(13) str9 pop00for90 %9s "Census 2000 population for 1990 census tract"
_column(22) str4 perc90 %4s "Percentage of 1990 census tract population"
_column(26) str2 state00 %2s "2000 state FIPS code"
_column(28) str3 county00 %3s "2000 county FIPS code"
_column(31) str4 tractbas00 %4s "2000 census tract base"

```

```
_column(35) str2 tractsuf00 %2s "2000 census tract suffix"
_column(37) str1 flag00 %1s "2000 census tract part flag"
_column(38) str9 pop00for00 %9s "Census 2000 population for 2000 census tract"
_column(47) str4 perc00 %4s "Percentage of 2000 census tract population"
_column(51) str9 areacov00 %9s "2000 population of the area covered by the record"
_column(60) str14 area00 %14s "Land area of the record (1000 sq. meters)"
_column(74) str2 stateabb %2s "2000 state name abbreviation"
_column(76) str60 countyname %60s "2000 county name"
}
```

Then I ran the following commands. Remember that we need a relationship file from 1990 to 2010, so we'll need to merge this newly imported file with the 2000-2010 relationship file previously created.

```
> destrstring , replace
> merge m:m state00 county00 tract00 using "E:\Research\Tract_relationship\2010\08014.dta"
> order state10 state00 state90 , last
> order county90 tract90 county00 tract00 county10 tract10 perc90 poppct00 , first
> drop if missing(county10)
> gen perc90to10 = perc90*poppct00/100
> order perc90to10 , before(pop00)
> sort county90 tract90 county00 tract00 county10 tract10
> keep county90 tract90 county10 tract10 perc90to10 state10
> gen FIPS90 = state10*1000+county90
> gen FIPS10 = state10*1000+county10
> save "E:\Research\Tract_relationship\2000\08014.dta"
```

Now, the 'perc90to10' field has information regarding the percentage of the population that was reassigned to Broomfield County, using 1990 census tract information. Now we can use this file to create Broomfield County from the data. Let's merge these two (CancerRisk1996\_mergeFIPS.dta & 2000\08014.dta); we'll use fields 'FIPS90' and 'tract90' as key fields. First we'll calculate the cancer risk for FIPS 08014.

```
> use E:\Research\EPA-NATA\1996\data\CancerRisk1996_mergeFIPS.dta
> rename tract tract90
> rename county countyname
> rename FIPS FIPS90
> drop _merge
> merge m:m FIPS90 tract90 using "E:\Research\Tract_relationship\2000\08014.dta"
> keep if FIPS10 == 8014
> replace population = population*perc90to10/100
> gen popindex = population*CancerRisk
> rename FIPS10 FIPS
> collapse (sum) popindex population , by(FIPS)
> gen CancerRisk = popindex/population*1000000
> drop popindex population
> save "E:\Research\EPA-NATA\1996\data\CancerRisk1996_08014_collapse.dta"
```

Now that the dta file for Broomfield County has been saved, we can work on the other counties. However, we need to work with the 2000\08014.dta before working with the other counties. This is because we need to create a dta file that has information regarding how many people were excluded from Broomfield County, not how many people were excluded from each census tract within Broomfield County. If we don't do this, we might have double-counting errors.

```
> use "E:\Research\Tract_relationship\2000\08014.dta", clear
> collapse (sum) perc90to10 , by(FIPS90 county90 tract90)
> replace perc90to10 = 100-perc90to10
> save "E:\Research\Tract_relationship\2000\outof08014.dta"
```

So, now this file (2000\outof08014.dta) has the percentage of population that remained outside of Broomfield County. We need to merge this file with the CancerRisk file previously created to calculate the risk on the other counties.

```
> use "E:\Research\EPA-NATA\1996\data\CancerRisk1996_mergeFIPS.dta", clear
> drop _merge
```

```
> merge m:m FIPS90 tract90 using "E:\Research\Tract_relationship\2000\outof08014.dta"
> replace perc90to10 = 100 if missing(perc90to10)
> replace population = population*perc90to10/100
> gen popindex = population*CancerRisk
> rename FIPS90 FIPS
> drop if FIPS==8014
> collapse (sum) popindex population , by(FIPS)
> gen CancerRisk = popindex/population*1000000
> drop popindex population
> save "E:\Research\EPA-NATA\1996\data\CancerRisk1996_but08014_collapse.dta"
```

Now that both files are saved, we can append them and then merge them with the FIPS.dta file.

```
> use E:\Research\EPA-NATA\1996\data\CancerRisk1996_but08014_collapse.dta
> append using "E:\Research\EPA-NATA\1996\data\CancerRisk1996_08014_collapse.dta"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> sort FIPS
> drop statecode _merge
> label variable CancerRisk "People at possible risk out of 1 million"
> save "E:\Research\EPA-NATA\1996\CancerRisk1996_collapse.dta"
```

### 43. Neurological risk

See Cancer risk.

### 44. Respiratory risk

See Cancer risk.

### 45. Local direct general expenditures

Local expenditure data is available for 2007. You will have to visit U.S. Census FTP site to access it at:

<http://www2.census.gov/pub/outgoing/govs/special60/>

Then click on \_IndFin\_1967-2007.zip. This file should contain all financial data for local governments. I suggest reviewing the following website <http://www.census.gov/govs/cog2012/> and if information is not available for 2017, contact the U.S. Census at [govs.finstaff@census.gov](mailto:govs.finstaff@census.gov).

Once we have downloaded the files we can import them into Stata and merge them.

```
> insheet using "E:\Research\Census-COG\data\_IndFin1967-2007\IndFin07a.Txt"
> save "E:\Research\Census-COG\COG2007a.dta"
> insheet using "E:\Research\Census-COG\data\_IndFin1967-2007\IndFin07b.Txt" , clear
> save "E:\Research\Census-COG\COG2007b.dta"
> insheet using "E:\Research\Census-COG\data\_IndFin1967-2007\IndFin07c.Txt" , clear
> save "E:\Research\Census-COG\COG2007c.dta"
> use "E:\Research\Census-COG\COG2007a.dta", clear
> merge 1:1 sortcode year4 id using "E:\Research\Census-COG\COG2007b.dta"
> drop _merge
> merge 1:1 sortcode year4 id using "E:\Research\Census-COG\COG2007c.dta"
> drop _merge
> save "E:\Research\Census-COG\COG2007.dta"
```

Now, since the dta file carries no FIPS code information, we need to extract this information from an excel file downloaded from the ftp site (GOVS\_to\_FIPS\_Codes\_State\_&\_County\_2007.xls). Open this file and keep the columns that represent the state ID, county ID, state FIPS code and county FIPS code, renaming these columns in this manner:

| Column name | Data stored |
|-------------|-------------|
| statecode   | state ID    |
| county      | county ID   |

|             |                  |
|-------------|------------------|
| FIPS_state  | state FIPS code  |
| FIPS_county | county FIPS code |

Once we have this excel file, we can save it and then import it into Stata, which we will use to create a FIPS variable in the COG2007.dta file, and then keep only the variables we need and collapse the information.

```
> import excel "E:\Research\Census-COG\data\_IndFin1967-2007\GOVS_to_FIPS_Codes.xls", sheet("FIPS") firstrow
clear
> destring , replace
> save "E:\Research\Census-COG\data\GOVs_to_FIPS.dta"
> use "E:\Research\Census-COG\COG2007.dta", clear
> merge m:m statecode county using "E:\Research\Census-COG\data\GOVs_to_FIPS.dta"
> gen FIPS = FIPS_state*1000+FIPS_county
> drop if _merge==2
> drop _merge
> save "E:\Research\Census-COG\COG2007.dta", clear
```

We need to keep the following variables:

|                     |   |
|---------------------|---|
| FIPS                | state/county FIPS code  |
| directgeneralexpend | Direct General Expenditure  |
| healthdirectexpend  | Health Direct Expenditure   |
| totalhospitaldirexp | Hospital Direct Expenditure   |
| naturalresdirectexp | Natural Resources Direct Expenditure                                  |
| parksrecredirectexp | Parks & Recreational Direct Expenditure                               |
| fedigrairtransport  | Federal intergovernmental revenue – Air Transportation                |
| fedigrhighways      | Federal intergovernmental revenue – Highways                          |
| fedigrtransitub     | Federal intergovernmental revenue – Transit subsidies                 |
| fedigrhouscomdev    | Federal intergovernmental revenue – Housing and community development |
| fedigrnaturalres    | Federal intergovernmental revenue – Natural Resources                 |
| fedigrsewerage      | Federal intergovernmental revenue – Sewerage                          |

Then using these variables, we can obtain the local direct general expenditure, local expenditure for hospitals and health, local expenditure on parks and natural resources, and federal expenditure.

```
> use E:\Research\Census-COG\COG2007.dta
> keep FIPS directgeneralexpend healthdirectexpend totalhospitaldirexp naturalresdirectexp parksrecredirectexp
fedigrairtransport fedigrhighways fedigrtransitub fedigrhouscomdev fedigrnaturalres fedigrsewerage
> order FIPS , first
> drop if FIPS-floor(FIPS/1000)*1000==0
> collapse (sum) directgeneralexpend healthdirectexpend totalhospitaldirexp naturalresdirectexp
parksrecredirectexp fedigrairtransport fedigrhighways fedigrtransitub fedigrhouscomdev fedigrnaturalres
fedigrsewerage , by(FIPS)
> drop if FIPS>1999&FIPS<3000
> drop if FIPS>14999&FIPS<16000
> drop if missing(FIPS)
> egen float FedExpend = rowtotal(fedigrairtransport fedigrhighways fedigrtransitub fedigrhouscomdev
fedigrnaturalres fedigrsewerage)
> drop fedigrairtransport fedigrhighways fedigrtransitub fedigrhouscomdev fedigrnaturalres fedigrsewerage
> egen float NatParkExpend = rowtotal(naturalresdirectexp parksrecredirectexp)
> drop naturalresdirectexp parksrecredirectexp
> egen float HospHealthExpend = rowtotal(healthdirectexpend totalhospitaldirexp)
> drop healthdirectexpend totalhospitaldirexp
> save "E:\Research\Census-COG\COG2007_collapse.dta"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode
```

There is no info for Bronx County (Bronx), Kings County (Brooklyn), Queens County (Queens), and Richmond County (Staten Island), within the State of New York. These expenditures are merged with the New York County (Manhattan). Therefore, I will consider the total population within these counties (Bronx, Kings,

Queens, Richmond, and New York) to calculate the per capita figure and then assume that this figure is the same for these counties.

```
> use "E:\Research\Census-COG\COG2007_collapse.dta", clear
> drop _merge
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> save "E:\Research\Census-COG\COG2007_collapse.dta", replace
> keep if FIPS==36005|FIPS==36047|FIPS==36081|FIPS==36085|FIPS==36061
> collapse (sum) population directgeneralexpend FedExpend NatParkExpend HospHealthExpend
> expand 5 in 1
> gen FIPS = 0
> replace FIPS = 36005 in 1
> replace FIPS = 36047 in 2
> replace FIPS = 36061 in 3
> replace FIPS = 36081 in 4
> replace FIPS = 36085 in 5
> append using "E:\Research\Census-COG\COG2007_collapse.dta"
> drop _merge
> drop if state=="New York"&county=="Bronx"
> drop if state=="New York"&county=="Kings"
> drop if state=="New York"&county=="Queens"
> drop if state=="New York"&county=="Richmond"
> drop if state=="New York"&county=="New York"
> drop county state
> order FIPS , first
> sort FIPS
> gen dirgen_localexp = directgeneralexpend/population
> label variable dirgen_localexp "Local direct general expenditure ($000 per capita)"
> gen hosphealth_localexp = HospHealthExpend/population
> label variable hosphealth_localexp "Local expenditure for hospitals and health ($000 per capita)"
> gen parknat_localexp = NatParkExpend/population
> label variable parknat_localexp "Local expenditure for parks, rec. and nat. resources ($000 per capita)"
> gen fedexp = FedExpend/population
> label variable fedexp "Federal expenditure ($000 per capita, non-wage, non-defense)"
> drop population directgeneralexpend FedExpend NatParkExpend HospHealthExpend
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> save "E:\Research\Census-COG\Local&Fed_Expenditure_COG2007.dta"
```

#### 46. Local exp. for hospitals and health

The same as 44. Local direct general expenditures

#### 47. Local exp. on parks, rec. and nat. resources

The same as 44. Local direct general expenditures

#### 48. Museums and historical sites

Data can be downloaded from the census website

([http://www.census.gov/econ/cbp/download/10\\_data/index.htm](http://www.census.gov/econ/cbp/download/10_data/index.htm)). Data is downloaded as .txt file and then imported into Stata. Museums and historical sites' NAICS code are equal to **71211** and **71212**, respectively.

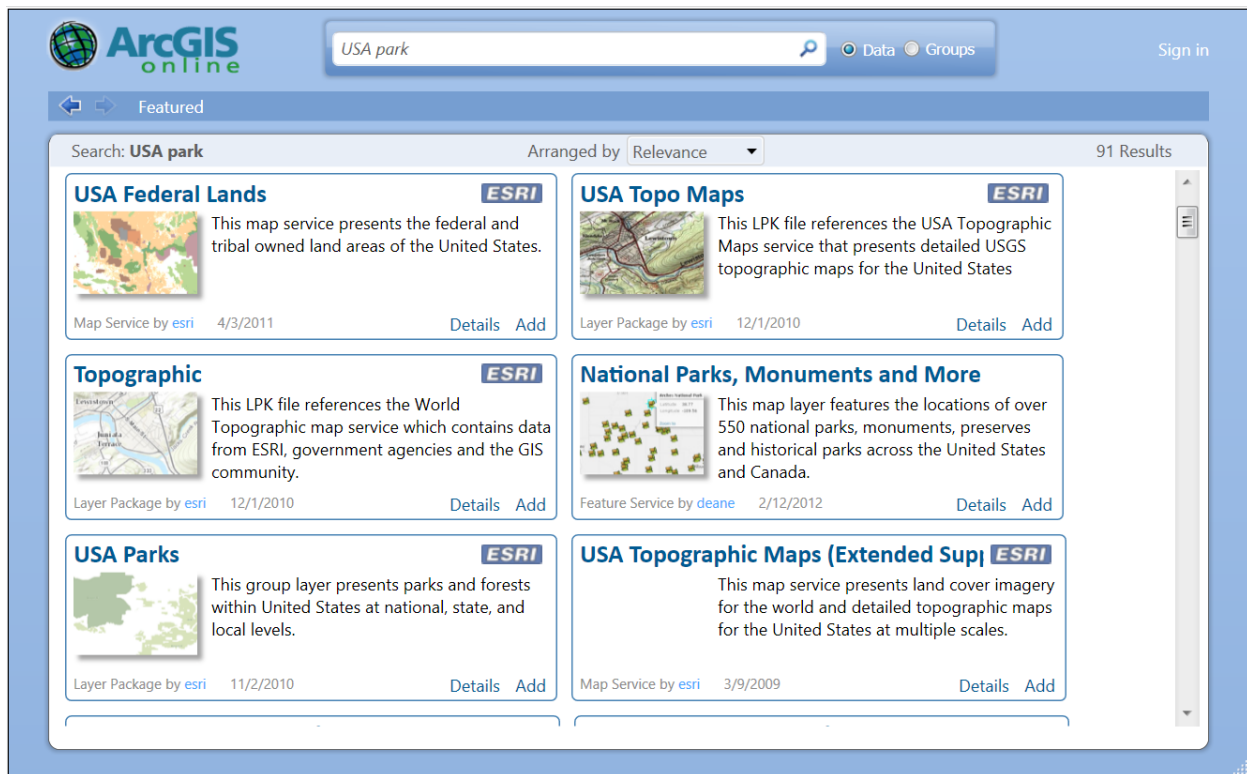
```
> insheet using "E:\Research\Census-CBP\data\cbp10co.txt", clear
> keep if naics=="71211/"|naics=="71212/"
> gen FIPS = fipstate*1000+fipscty
> keep FIPS est
> collapse (sum) est , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if FIPS>2000&FIPS<3000
> drop if FIPS>15000&FIPS<16000
> sort FIPS
> replace est = 0 if missing(est)
```



```
> rename est museums
> label variable museums "Number of museums and historical sites"
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> drop if _merge==1
> drop _merge
> gen museums_per1000 = museums/population*1000
> label variable museums_per1000 "Number of museums and historical sites per 1000 people"
> order state county FIPS , first
> order population , last
> save "E:\Research\Census-CBP\Museums_2010_dhm.dta"
```

## 49. Municipal parks (percentage of total land area)

Download data from ArcGIS Online. Select 'Add Data' and then 'Add Data from ArcGIS Online...' in ArcMap. Search USA Park and then select 'USA Parks'.



The data will have 4 groups, categorized by scale. Degrup the data and then just keep the last group, the one with the biggest scale. Export this data to save this shapefile. This shapefile will have USA parks, including federal, state, county and local parks. Since we are only interested in local parks, we need to open the attribute table and then look for the field that stores this information. In this case, the field 'Feattype' has several categories; one of them is 'Local Park'. Select by attributes and then export the selected data into a different shapefile.

Then we need to project this into a suitable projected coordinate system. Since we are going to calculate areas, I decided to project the shapefile to the 'USA Contiguous Albers Equal Area Conic' projected coordinate system.

We need to run the 'Intersect' tool (ArcToolbox \ Analysis Tools \ Overlay \ Intersect), between the local park shapefile and the county land shapefile. Then select the following options:

- (1) Input features: Dissolved Federal area shapefile & county land shapefile.
- (2) Output feature class: Path and name of the new shapefile. This shapefile will have just the federal land.

- (3) Then, we need to open the attribute tables of the newly created shapefile.
- (4) Add a new field named: 'fed\_mile2' and then right click on it and select calculate geometry. Calculate the area in square miles.
- (5) Export the attribute table as a txt file.

Then we can import this table into Stata and use the information to calculate the federal land as percentage of the total land of the county. Run the following commands to import these txt files into Stata and format the information:

```
> insheet using "E:\Research\NationalAtlas\LocalPark\LocalParkCounty.txt"
> gen FIPS = statefp10*1000 + countyfp10
> collapse (sum) localmi2 , by(FIPS)
> rename localmi2 localpark_mi2
> label variable localpark_mi2 "Local park land in square miles"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> sort FIPS
> drop _merge statecode
> replace localpark_mi2 = 0 if missing(localpark_mi2)
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyLand_2010.dta"
> drop _merge
> gen localpark_pct = localpark_mi2/land_mi2*100
> label variable localpark_pct "Local park land in percentage (over land area)"
> order FIPS state county , first
> drop land_mi2
> save "E:\Research\NationalAtlas\LocalPark\LocalPark2010.dta"
```

## 50. Campgrounds and camps

Data can be downloaded from the census website

([http://www.census.gov/econ/cbp/download/10\\_data/index.htm](http://www.census.gov/econ/cbp/download/10_data/index.htm)). Data is downloaded as .txt file and then imported into Stata. Campgrounds and camps have the following NAICS codes (2007):

72121 Recreational vehicle parks and campgrounds

72124 Recreational and vacation camps

```
> insheet using "E:\Research\Census-CBP\data\cbp10co.txt", clear
> keep if naics=="72121/"|naics=="72124/"
> gen FIPS = fipstate*1000+fipscty
> keep FIPS est
> collapse (sum) est , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if FIPS>2000&FIPS<3000
> drop if FIPS>15000&FIPS<16000
> sort FIPS
> replace est = 0 if missing(est)
> rename est camps
> label variable camps "Number of campgrounds and camps"
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> drop if _merge==1
> drop _merge
> gen camps_per1000 = camps/population*1000
> label variable camps_per1000 "Number of campgrounds and camps per 1000 people"
> order state county FIPS , first
> order population , last
> save "E:\Research\Census-CBP\Camps_2010_dhm.dta"
```

## 51. Zoos, botanical gardens and nature parks

Data can be downloaded from the census website

([http://www.census.gov/econ/cbp/download/10\\_data/index.htm](http://www.census.gov/econ/cbp/download/10_data/index.htm)). Data is downloaded as .txt file and then imported into Stata. Zoos, botanical gardens and nature parks have the following NAICS codes (2007):

71213 Zoos and botanical gardens

## 71219 Nature parks and other similar institutions

```
> insheet using "E:\Research\Census-CBP\data\cbp10co.txt", clear
> keep if naics=="71213/"|naics=="71219/"
> gen FIPS = fipstate*1000+fipscty
> keep FIPS est
> collapse (sum) est , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if FIPS>2000&FIPS<3000
> drop if FIPS>15000&FIPS<16000
> sort FIPS
> replace est = 0 if missing(est)
> rename est zoos
> label variable zoos "Number of zoos, botanical gardens and nature parks"
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> drop if _merge==1
> drop _merge
> gen zoos_per1000 = zoos/population*1000
> label variable zoos_per1000 "Number of zoos, botanical gardens and nature parks per 1000 people"
> order state county FIPS , first
> order population , last
> save "E:\Research\Census-CBP\Zoos_2010_dhm.dta"
```

## 52. Crime rate (per 100,000 persons)

Crime here refers to Personal crime, not property crime.

See introduction in <http://www.icpsr.umich.edu/icpsrweb/content/NACJD/guides/ucr.html>

Data is accessible from ICPSR at

<http://www.icpsr.umich.edu/icpsrweb/NACJD/studies/30763?q=Uniform+Crime+Reporting+Program+Data+%5BUnited+States%5D%3A+County-Level&archive=NACJD&y=13&x=29&sortBy=5&paging.rows=25>.

STATA format not available.

Data availability: yr 2008 (2010s), 2000 (2000s), 1990 (1990s), 1984 (1980s)

The latest data is in 2008. Raw data is stored as [C:\Data\ICPSR\ICPSR\\_2008](#). Using dictionary (write using the codebook), data is read into STATA. Population in the same year was collected and saved in [C:\Data\ICPSR\Population](#). The crime rate is calculated as "P1VLNT/(population/100,000)."

Raw crime data for 2000, 1990, 1984, and 1970 are stored in [C:\Data\ICPSR\ICPSR\\_2000](#), [C:\Data\ICPSR\ICPSR\\_1990](#), [C:\Data\ICPSR\ICPSR\\_1984](#), and [C:\Data\ICPSR\ICPSR\\_1970](#)\_need to interpret.

Parts 1-3: Codebook for UCR County-Level Arrests Data  
Parts 5-7: Codebook for Allocated Statewide Data for Arrests  
Part 4: Codebook for UCR County-Level Crimes Reported Data  
Part 8: Codebook for Allocated Statewide Data for Crimes Reported

ALSO: not use NCVS for county crime rate.

**NOTICE: not pass the quality control test**

**The reason could be the variables:**

**"SP1VLNT PART 1-VIOLENT CRIMES 56-60 5**

**Sum of variables SMURDER**

**through SAGASSLT." Not included in.**

### 53. Teacher-pupil ratio

Data can be downloaded from the Census of Governments Database. This data can be found in the U.S. Census FTP site:

<http://www2.census.gov/pub/outgoing/govs/special60/>

Then click on `_IndEmp_1967-2010.zip`. This file should contain all employment data for local governments. I suggest reviewing the following website <http://www.census.gov/govs/cog2012/> and if information is not available for 2020, then contact the U.S. Census at [govs.finstaff@census.gov](mailto:govs.finstaff@census.gov).

Once we have downloaded and unzipped the zip file we can import it into Stata. I tried to use the 2009 data, but it is not complete (it only contains information for less than 2600 counties); the 2007 is complete, so I decided to use it. Also, I am only considering full- and part-time instructional employees from elementary and secondary education institutions (Emp. Code 012) as 'teachers'.

```
> insheet using "E:\Research\Census-COG\data\_IndEmp1972-2010\IndEmp07.Txt", clear
> save "E:\Research\Census-COG\COG_Emp_2007.dta"
> import excel "E:\Research\Census-COG\data\_IndEmp1972-2010\GOVS_to_FIPS.xls", sheet("FIPS") firstrow clear
> destring , replace
> save "E:\Research\Census-COG\data\_IndEmp1972-2010\GOVS_to_FIPS.dta"
> use "E:\Research\Census-COG\COG_Emp_2007.dta", clear
> merge m:m statecode county using "E:\Research\Census-COG\data\_IndEmp1972-2010\GOVS_to_FIPS.dta"
> gen FIPS = FIPS_state*1000+FIPS_county
> drop if _merge==2
> drop _merge
> save "E:\Research\Census-COG\COG_Emp_2007.dta", replace
> keep FIPS elemeducinstrtotalemp
> order FIPS , first
> drop if FIPS-floor(FIPS/1000)*1000==0
> collapse (sum) elemeducinstrtotalemp , by(FIPS)
> drop if FIPS>1999&FIPS<3000
> drop if FIPS>14999&FIPS<16000
> drop if missing(FIPS)
> save "E:\Research\Census-COG\COG2007_emp_collapse.dta"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> merge 1:1 FIPS using "E:\Research\Census-ACS\Students_2010.dta"
> drop _merge
> save "E:\Research\Census-COG\COG2007_emp_collapse.dta", replace

> keep if FIPS==36005|FIPS==36047|FIPS==36081|FIPS==36085|FIPS==36061
> collapse (sum) students elemeducinstrtotalemp
> expand 5 in 1
> gen FIPS = 0
> replace FIPS = 36005 in 1
> replace FIPS = 36047 in 2
> replace FIPS = 36061 in 3
> replace FIPS = 36081 in 4
> replace FIPS = 36085 in 5
> append using "E:\Research\Census-COG\COG2007_emp_collapse.dta"
> drop _merge
> drop if state=="New York"&county=="Bronx"
> drop if state=="New York"&county=="Kings"
> drop if state=="New York"&county=="Queens"
> drop if state=="New York"&county=="Richmond"
> drop if state=="New York"&county=="New York"
> drop county state total undergrad grad
> order FIPS , first
> sort FIPS
> gen teacher_pupil_ratio = elemeducinstrtotalemp/students
> label variable teacher_pupil_ratio "Teacher-pupil ratio"
> drop students elemeducinstrtotalemp
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> order FIPS state county , first
> save "E:\Research\Census-COG\Teacher_Pupil_ratio_COG2007.dta"
```

## 54. Local expenditure per student

Since the Local expenditure per student is not calculated using the whole county population, the procedure is similar, though not the same, to 'Local direct general expenditure', so I decided to describe the whole specific procedure in this section. I am using the COG2007.dta file created in 'Local direct general expenditure'.

```
> use "E:\Research\Census-COG\COG2007.dta", clear
> keep FIPS elemeducdirectexp
> order FIPS , first
> collapse (sum) elemeducdirectexp , by(FIPS)
> drop if missing(FIPS)
> drop if FIPS>1999&FIPS<3000
> drop if FIPS>14999&FIPS<16000
> drop if FIPS-floor(FIPS/1000)*1000==0
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> label variable elemeducdirectexp "Elementary and High School direct education expenditure in $000"
> drop statecode _merge
> save "E:\Research\Census-COG\COG2007_edu_collapse.dta"
```

Again, there is no expenditure info for Bronx County (Bronx), Kings County (Brooklyn), Queens County (Queens), and Richmond County (Staten Island), within the State of New York. These expenditures are merged with the New York County (Manhattan). Therefore, I will consider the total enrolled students within these counties (Bronx, Kings, Queens, Richmond, and New York) to calculate the per student figure and then assume that this figure is the same for these counties.

Now that we have the total direct education expenditure at the county level we need the number of students in each county. This data can be gathered from the 2009 ACS 5-year estimate. Go to the Advanced Search of the American FactFinder website (<http://factfinder2.census.gov/faces/nav/jsf/pages/searchresults.xhtml?refresh=t>).

Then select the following:

- Topics\People\Education\School Enrollment
- Topics\Dataset\2009 ACS 5-year estimates
- Geographies:
  - Select from: most requested geographic types
  - Select a geographic type: ...County – 050
  - Select one or more geographic areas and click Add to Your Selections: All Counties within the United States.
  - Click 'Add to your Selections'

Then click the first table from the Search Results, it should be the School Enrollment table (ID: S1401). You will be prompted to the 'Table view', where you can click on 'Download' and download the table as a .csv file. The .csv file contains school enrollment information from preschool until graduate school, so we need to subtract undergraduate and graduate figures from the enrollment number.

```
> insheet using "E:\Research\Census-ACS\data\2009ACS-5year\ACS_09_5YR_S1401_with_ann.csv", clear
> rename geoid2 FIPS
> rename hc01_est_vc01 total
> rename hc01_est_vc08 undergrad
> rename hc01_est_vc09 grad
> keep FIPS total undergrad grad
> label variable FIPS ""
> label variable total "Total enrolled students"
> label variable undergrad "Undergraduate students"
> label variable grad "Graduate students"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==1
> drop statecode _merge
> gen students = total-undergrad-grad
> label variable students "Preschool and school students"
> save "E:\Research\Census-ACS\Students_2010.dta"
```

Now we can merge the two dta files together and calculate the expenditure per student figure. Remember that there is no expenditure info for Bronx County (Bronx), Kings County (Brooklyn), Queens County (Queens), and Richmond County (Staten Island), within the State of New York.

```
> use "E:\Research\Census-COG\COG2007_edu_collapse.dta", clear
> merge 1:1 FIPS using "E:\Research\Census-ACS\Students_2010.dta"
> save "E:\Research\Census-COG\COG2007_edu_collapse.dta", replace
> keep if FIPS==36005|FIPS==36047|FIPS==36081|FIPS==36085|FIPS==36061
> collapse (sum) students elemeducdirectexp
> expand 5 in 1
> gen FIPS = 0
> replace FIPS = 36005 in 1
> replace FIPS = 36047 in 2
> replace FIPS = 36061 in 3
> replace FIPS = 36081 in 4
> replace FIPS = 36085 in 5
> append using "E:\Research\Census-COG\COG2007_edu_collapse.dta"
> drop _merge
> drop if state=="New York"&county=="Bronx"
> drop if state=="New York"&county=="Kings"
> drop if state=="New York"&county=="Queens"
> drop if state=="New York"&county=="Richmond"
> drop if state=="New York"&county=="New York"
> drop county state total undergrad grad
> order FIPS , first
> sort FIPS
> gen edu_localexp = elemeducdirectexp/students
> label variable edu_localexp "Local education expenditure ($000 per student)"
> drop students elemeducdirectexp
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> order FIPS state county , first
> save "E:\Research\Census-COG\Edu_Expenditure_COG2007.dta"
```

## 55. Private school to public school enrollment (%)

Data comes from the 2009 ACS 5-year estimate. See second part of 'Local expenditure per student' to see instructions on how to download the school enrollment .csv file. Then we can run the following commands to format the dta file:

```
> insheet using "E:\Research\Census-ACS\data\2009ACS-5year\ACS_09_5YR_S1401_with_ann.csv", clear
> rename geoid2 FIPS
> rename hc01_est_vc01 total
> rename hc03_est_vc01 totalpub_percent
> rename hc04_est_vc01 totalpriv_percent
> rename hc01_est_vc08 undergrad
> rename hc03_est_vc08 undergradpub_percent
> rename hc04_est_vc08 undergradpriv_percent
> rename hc01_est_vc09 grad
> rename hc03_est_vc09 gradpub_percent
> rename hc04_est_vc09 gradpriv_percent
> keep FIPS total totalpub_percent totalpriv_percent undergrad undergradpub_percent undergradpriv_percent grad
gradpub_percent gradpriv_percent
> replace undergradpub_percent = "0" if undergradpub_percent=="-"
> replace undergradpriv_percent = "0" if undergradpriv_percent=="-"
> replace gradpub_percent = "0" if gradpub_percent=="-"
> replace gradpriv_percent = "0" if gradpriv_percent=="-"gen totalpub = totalpub_percent*total/100
> destring , replace
> gen totalpub = totalpub_percent*total/100
> gen totalpriv = totalpriv_percent*total/100
> gen undergradpub = undergradpub_percent*undergrad/100
> gen undergradpriv = undergradpriv_percent*undergrad/100
> gen gradpub = gradpub_percent*grad/100
> gen gradpriv = gradpriv_percent*grad/100
> gen studentspub = totalpub-undergradpub-gradpub
> gen studentspriv = totalpriv-undergradpriv-gradpriv
```

```
> replace studentspub = 0 if studentspub<0
> replace studentspriv = 0 if studentspriv<0
> gen priv_to_pub = studentspriv/studentspub*100
## Note: There is one missing value in priv_to_pub. There are 0 students enrolled in a public school and 0
students enrolled in a private school in this county, so I decided to replace the missing value with a 100
(Since the number of students enrolled in a private school and the number of students enrolled in a public
school are the same, i.e. 0).
> replace priv_to_pub = 0 if missing(priv_to_pub)
> label variable priv_to_pub "Private school to public school enrollment (%)"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==1
> keep FIPS state county priv_to_pub
> order FIPS state county , first
> label variable FIPS ""
> save "E:\Research\Census-ACS\Priv_to_Pub_2010.dta"
```

## 56. Child mortality (per 1000 births, 1990–2000)

Data can be downloaded from the CDC Wonder website (<http://wonder.cdc.gov/lbd.html>). By selecting each record hyperlink, one can select also specific years so we can also work with a dataset from 2000 to 2009, even though we have 1999-2002, 2003-2006, and 2007-2009 datasets. After clicking on one of the records, select the following:

1. Organize table layout: group results by: county.
2. Select maternal residence: States, Select ALL states.
3. Select all other maternal characteristics: Select ALL in every category.
4. Select birth characteristics: Select ALL in every category.
5. Select cause of infant death: Select ICD-10 Codes, Select ALL causes of death.
6. Select infant characteristics: Select All ages, All years, All genders. (When working with the 1999-2002 dataset, select 2000, 2001 and 2002 years).
7. Other options: Check export results, show totals, show zero values, show suppressed values, precision: 2, data access timeout: 5 minutes, calculate rates per: 1,000.

Once we have exported the txt file we are ready to import them into Stata and format the datasets. We can run the following commands to do so:

```
> insheet using "E:\Research\CDC-ChildMortality\data\Linked Birth Infant Death Records, 2000-2002.txt", tab
clear
> rename countycode FIPS
> rename county countyST
> drop notes
> drop if missing(FIPS)
> drop if FIPS<3000&FIPS>2000
> drop if FIPS<16000&FIPS>15000
> replace FIPS=12086 if FIPS==12025
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> sort FIPS
> drop state county statecode _merge countyST
> save "E:\Research\CDC-ChildMortality\ChildM2000_2002.dta"

> insheet using "E:\Research\CDC-ChildMortality\data\Linked Birth Infant Death Records, 2003-2006.txt", tab
clear
> rename countycode FIPS
> rename county countyST
> drop notes
> drop if missing(FIPS)
> drop if FIPS<3000&FIPS>2000
> drop if FIPS<16000&FIPS>15000
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> sort FIPS
> drop state county statecode _merge drop countyST
> save "E:\Research\CDC-ChildMortality\ChildM2003_2006.dta"
```



```

> insheet using "E:\Research\CDC-ChildMortality\data\Linked Birth Infant Death Records, 2007-2009.txt", tab
clear
> rename countycode FIPS
> rename county countyST
> drop notes
> drop if missing(FIPS)
> drop if FIPS<3000&FIPS>2000
> drop if FIPS<16000&FIPS>15000
> replace FIPS=12086 if FIPS==12025
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> sort FIPS
> drop state county statecode _merge countyST
> save "E:\Research\CDC-ChildMortality\ChildM2007_2009.dta"

> use "E:\Research\CDC-ChildMortality\ChildM2000_2002.dta", clear
> append using "E:\Research\CDC-ChildMortality\ChildM2003_2006.dta" "E:\Research\CDC-
ChildMortality\ChildM2007_2009.dta"
> collapse (sum) deaths births , by (FIPS)
> gen childmort = deaths/births*1000
> label variable childmort "Child mortality rate (deaths per 1000 births), 2000-2009"
> gen statecode = floor(FIPS/1000)
> gen countycode = FIPS-statecode*1000
> replace childmort = childmort(countycode==999)
> gen childmort999 = 0
> replace childmort999 = childmort if countycode==999
> save "E:\Research\CDC-ChildMortality\ChildM2000_2009.dta"

> keep if countycode==999
> keep statecode childmort999
> save "E:\Research\CDC-ChildMortality\childmort999.dta"

> use "E:\Research\CDC-ChildMortality\ChildM2000_2009.dta", clear
> drop childmort999
> merge m:1 statecode using "E:\Research\CDC-ChildMortality\childmort999.dta"
> browse if _merge==1
> drop _merge
> replace childmort = childmort999 if missing(childmort)
> gen Notes = "Generated from unidentified counties mean due to lack of information"
> replace Notes = "" if births!=0
> drop deaths births statecode countycode childmort999
> merge m:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop if _merge==1
> sort FIPS
> drop _merge statecode
> order FIPS state county , first
> save "E:\Research\CDC-ChildMortality\ChildM2010.dta"

```

## 57. Federal expenditure (\$ pc, non-wage, non-defense)

I am considering 'Federal expenditure' as Intergovernmental revenue from the Federal Government which is dedicated to Air Transportation, Highways, Transit subsidies, Housing and community development, Natural Resources and Sewerage. The list of expenditure categories within the Intergovernmental Revenue from Federal Government is the following:

| Category  | Taken into account |
|---|--------------------|
| Air Transportation (airports)                         | Yes                |
| Education   | No                 |
| Employment security (social insurance) administration | No                 |
| General support                                       | No                 |
| General revenue sharing                               | No                 |
| Other general support                                 | No                 |
| Health and hospitals                                  | No                 |
| Highways  | Yes                |



|                                   |     |
|-----------------------------------|-----|
| Transit subsidies                 | Yes |
| Housing and community development | Yes |
| Natural resources                 | Yes |
| Public welfare                    | No  |
| Sewerage                          | Yes |
| Others                            | No  |

These variables have been 'kept' from the 2007 COG dataset. See 'Local Direct General Expenditures'.

## 58. Number of Airports

The National Transportation Atlas Database (NTAD) 2010 from the Research and Innovative Technology Administration – Bureau of Transportation Statistics can be ordered from the following website:  
<https://1bts.rita.dot.gov/pdc/user/products/src/products.xml?p=3194&c=-1>

However, this data is only available in DVD, so you will have to order it.

Airports 2010 data is found in the National Transportation Atlas Database 2010, so we need to order it. We need to load the shapefile into ArcMap and then join this shapefile with the county boundary shapefile. In order to do this we need to do the following:

1. Right click the County boundary layer.
2. Select 'Joins and Relates \ Join ...'
  - a. What do you want to join to this layer? Join data from another layer based on spatial location
  - b. Choose the layer to join to this layer, or load spatial data from disk: Airports
  - c. Each polygon (i.e. county) will be given a count field of the airport layer, therefore counting how many airports each county has.
  - d. Specify output shapefile or feature class for this new layer:  
 E:\Research\GIS\Shapefiles\Transportation\NAD83\Airports2010\_county.shp
3. Open the attribute table of the newly created shapefile.
4. Export the attribute table as a txt file.

Then we need to import this txt file into Stata and collapse the data in order to obtain the number of airports per county. Run the following commands.

```
> insheet using "E:\Research\USDT-NTAD\data\NTAD2010\point\airports\airports_county.txt", clear
> keep statefp10 countyfp10 count_
> rename count_ airports
> gen FIPS = statefp10*1000+countyfp10
> order FIPS , first
> keep FIPS airports
> label variable airports "Number of airports, NTAD 2010"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop _merge statecode
> save "E:\Research\USDT-NTAD\Airports\airports_2010.dta"
```

## 59. Number of Ports

Ports 2010 data can be found in the National Transportation Atlas Database **2011**

([http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national\\_transportation\\_atlas\\_database/index.html](http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_atlas_database/index.html)).

After downloading the shapefile it is important to read the txt file. This file describes the fields that the attribute table has. Since there are county and state fields, there would be no need to project this information and we should be able to use the dbf file directly to gather ports location information. However, after doing this, several ports had erroneous location information. For instance, some said that were located on the right bank of a river but also said that they were located on the state that was on the left bank. In order to reduce errors, I decided to plot the ports and then work with their locations in ArcGIS. We need to load it into ArcMap and then join this shapefile with the county boundary shapefile. In order to do this we need to do the following:

1. Right click the County boundary layer.
2. Select 'Joins and Relates \ Join ...'
  - a. What do you want to join to this layer? Join data from another layer based on spatial location
  - b. Choose the layer to join to this layer, or load spatial data from disk: Ports
  - c. Each polygon (i.e. county) will be given a count field of the airport layer, therefore counting how many airports each county has.
  - d. Specify output shapefile or feature class for this new layer:  
E:\Research\GIS\Shapefiles\Transportation\NAD83\Ports2010\_county.shp
3. Open the attribute table of the newly created shapefile.
4. Export the attribute table as a txt file.

Then we need to import this txt file into Stata and collapse the data in order to obtain the number of ports per county. Run the following commands.

```
> insheet using "E:\Research\USDT-NTAD\data\NTAD2011\point\ports\ports_county.txt", comma clear
> keep statefp10 countyfp10 count_
> rename count_ ports
> gen FIPS = statefp10*1000+countyfp10
> order FIPS , first
> keep FIPS ports
> label variable ports "Number of ports, NTAD 2011, data 2010"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop _merge statecode
> save "E:\Research\USDT-NTAD\Ports\ports_2010.dta", replace
```

## 60. Interstate highways (total mileage per mi<sup>2</sup>)

The National Transportation Atlas Database (NTAD) 2010 from the Research and Innovative Technology Administration – Bureau of Transportation Statistics can be ordered from the following website:

<https://1bts.rita.dot.gov/pdc/user/products/src/products.xml?p=3194&c=-1>

However, this data is only available in DVD, so you will have to order it.

Interstate highway data can be found in the 'nhpn' shapefile (nhpn stands for National Highway Planning Network). We need to load the shapefile into ArcMap and keep the lines that represent interstate highways. We can do that by selecting by attributes and then choosing the rows that have SINGT = 'I' (Read the txt file in order to choose which field to select and then which value in that field). Now that we have a shapefile with just the Interstate highways within the Contiguous U.S., we can use this shapefile to obtain the total mileage of these highways within each county. In order to do this we need to do the following:

1. Project the Interstate highways shapefile to a suitable projection. Since we are going to calculate distances, we need to use the 'USA Contiguous Equidistant Conic' projection. Run the Project tool (ArcToolbox \ Data Management Tools \ Projections and Transformations \ Feature \ Project).
  - a. Input Dataset or feature class: Interstate highway shapefile
  - b. Output Dataset or Feature Class:  
E:\Research\GIS\Shapefiles\Transportation\Equidistant\Interstate\_hw2010.shp
  - c. Output Coordinate System: USA Contiguous Equidistant Conic
2. Run the Intersect tool (ArcToolbox \ Analysis Tools \ Overlay \ Intersect). This tool will split the Interstate Highway shapefile using the county boundaries; therefore we will obtain highways that span only within the county.
  - a. Input features: both shapefiles (Interstate highways & County boundaries)
  - b. Output feature class:  
E:\Research\GIS\Shapefiles\Transportation\Equidistant\Interstate\_hw2010\_county.shp
3. Open the attribute field of the newly created shapefile.
4. Add a new field, name it 'length\_mi'.
5. Right click this new field and click 'Calculate geometry'. Then select length in miles [US].
6. Export the attribute table as a txt file.

Then we need to import this txt file into Stata and collapse the data in order to obtain the mileage of Interstate highways per square miles of land per county. Run the following commands.

```
> insheet using "E:\Research\USDT-NTAD\data\NTAD2010\polyline\nhpn\interstate_hw_county.txt"
> keep statefp10 countyfp10 length_mi
> gen FIPS = statefp10*1000+countyfp10
> keep FIPS length_mi
> order FIPS , first
> rename length_mi InterstateHwys
> collapse (sum) InterstateHwys , by(FIPS)
> label variable InterstateHwys "Mileage of Interstate Highways per county, NTAD 2010"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop _merge statecode
> replace InterstateHwys = 0 if missing(InterstateHwys)
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyLand_2010.dta"
> drop _merge
> gen InterstateHwys_perland = InterstateHwys/land_mi2
> drop land_mi2
> label variable InterstateHwys_perland "Mileage of Interstate Highways per county (mi/mi2 of land), NTAD 2010"
> save "E:\Research\USDT-NTAD\InterstateHighways\InterstateHighways_2010.dta"
```

## 61. Urban arterial (total milage per mi<sup>2</sup>)

See Section 59 Interstate highways (total mileage per mi<sup>2</sup>); the procedure is almost the same. However, in this case we need to select the observations that have FCLASS = 14 or FCLASS = 16 (Read the txt file in order to choose which field to select and then which value in that field)<sup>8</sup>. Then run the following commands in Stata.

```
> insheet using "E:\Research\USDT-NTAD\data\NTAD2010\polyline\nhpn\urban_arterials_county.txt", clear
> keep statefp10 countyfp10 length_mi
> gen FIPS = statefp10*1000+countyfp10
> keep FIPS length_mi
> order FIPS , first
```

<sup>8</sup> FCLASS = 11: Urban Principal Arterial – Interstate

FCLASS = 12 Urban Principal Arterial – Other Freeways & Expressways

FCLASS = 14: Urban Principal Arterial – Other

FCLASS = 16: Urban Minor - Arterial

```
> rename length_mi UrbanArterials
> collapse (sum) UrbanArterials , by(FIPS)
> label variable UrbanArterials "Mileage of urban arterials per county, NTAD 2010"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop _merge statecode
> replace UrbanArterials = 0 if missing(UrbanArterials)
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyLand_2010.dta"
> drop _merge
> gen UrbanArterials_perland = UrbanArterials/land_mi2
> drop land_mi2
> label variable UrbanArterials_perland "Mileage of urban arterials per county (mi/mi2 of land), NTAD 2010"
> save "E:\Research\USDT-NTAD\UrbanArterials\UrbanArterials_2010.dta"
```

## 62. Number of Amtrak stations

Amtrak stations 2010 data can be found in the National Transportation Atlas Database 2011

([http://www.rita.dot.gov/bts/sites/rita.dot.gov/bts/files/publications/national\\_transportation\\_atlas\\_database/index.html](http://www.rita.dot.gov/bts/sites/rita.dot.gov/bts/files/publications/national_transportation_atlas_database/index.html)).

See Section 57 Number of Airports for working with the Amtrak shapefile, and then run the following commands in Stata.

```
> insheet using "E:\Research\USDT-NTAD\data\NTAD2011\point\amtrk_sta\amtrk_county.txt", comma clear
> keep statefp10 countyfp10 count_
> rename count_ amtrak
> gen FIPS = statefp10*1000+countyfp10
> order FIPS , first
> keep FIPS amtrak
> label variable amtrak "Number of Amtrak stations, NTAD 2011, data 2010"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop _merge statecode
> save "E:\Research\USDT-NTAD\Amtrak\amtrak_2010.dta", replace
```

## 63. Number of urban rail stops

The data is accessible from <http://nationalatlas.gov/atlasftp.html?openChapters=chptrans#chptrans>. The latest is in 2005. Data is stored in [C:\Data\Railroad](#).

## 64. Railways (total mileage per mi<sup>2</sup>)

See Section 59 Interstate highways (total mileage per mi<sup>2</sup>); the procedure is almost the same. Then run the following commands in Stata.

```
> insheet using "E:\Research\USDT-NTAD\data\NTAD2010\polyline\nhpn\urban_arterials_county.txt", clear
> keep statefp10 countyfp10 length_mi
> gen FIPS = statefp10*1000+countyfp10
> keep FIPS length_mi
> order FIPS , first
> rename length_mi Railways
> collapse (sum) Railways , by(FIPS)
> label variable Railways "Mileage of railways per county"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop _merge statecode
> replace Railways = 0 if missing(Railways)
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyLand_2010.dta"
> drop _merge
> gen Railways_perland = Railways/land_mi2
> drop land_mi2
> label variable Railways_perland "Mileage of railways per county (mi/mi2 of land), NTAD 2010"
> save "E:\Research\USDT-NTAD\Railways\Railways_2010.dta"
```

## 65. Number of restaurants and bars (per 1,000 people)

Data can be downloaded from the census website

([http://www.census.gov/econ/cbp/download/10\\_data/index.htm](http://www.census.gov/econ/cbp/download/10_data/index.htm)). Data is downloaded as .txt file and then imported into Stata. Restaurant and bars have the following NAICS codes (2007):

722110 Full-service restaurants  
722410 Drinking places (alcoholic beverages)

```
> insheet using "E:\Research\Census-CBP\data\cbp10co.txt", clear
> keep if naics=="722110"|naics=="722410"
> gen FIPS = fipstate*1000+fipscty
> keep FIPS est
> collapse (sum) est , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if FIPS>2000&FIPS<3000
> drop if FIPS>15000&FIPS<16000
> sort FIPS
> replace est = 0 if missing(est)
> rename est restau_bar
> label variable restau_bar "Number of restaurants and bars"
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> drop if _merge==1
> drop _merge
> gen restau_bar_per1000 = restau_bar/population*1000
> label variable restau_bar_per1000 "Number of restaurants and bars per 1000 people"
> order FIPS state county , first
> order population , last
> save "E:\Research\Census-CBP\Restaurants_2010_dhm.dta"
```

## 66. Theatres and musicals (per 1,000 people)

Data can be downloaded from the census website

([http://www.census.gov/econ/cbp/download/10\\_data/index.htm](http://www.census.gov/econ/cbp/download/10_data/index.htm)). Data is downloaded as .txt file and then imported into Stata. Theatres and musicals have the following NAICS codes (2007):

711110 Theatre Companies and Dance Theatres  
711120 Dance companies  
711130 Musical group and artists  
711190 Other performing arts companies

```
> insheet using "E:\Research\Census-CBP\data\cbp10co.txt", clear
> keep if naics=="711110"|naics=="711120"|naics=="711130"|naics=="711190"
> gen FIPS = fipstate*1000+fipscty
> keep FIPS est
> collapse (sum) est , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if FIPS>2000&FIPS<3000
> drop if FIPS>15000&FIPS<16000
> sort FIPS
> replace est = 0 if missing(est)
> rename est theatre
> label variable theatre "Number of theatres and musicals"
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> drop if _merge==1
> drop _merge
> gen theatre_per1000 = theatre/population*1000
> label variable theatre_per1000 "Number of theatres and musicals per 1000 people"
> order FIPS state county , first
> order population , last
> save "E:\Research\Census-CBP\Theatre_2010_dhm.dta"
```

## 67. Artists (per 1,000 people)

Data can be downloaded from the census website

([http://www.census.gov/econ/cbp/download/10\\_data/index.htm](http://www.census.gov/econ/cbp/download/10_data/index.htm)). Data is downloaded as .txt file and then imported into Stata. 'Artists' has the following NAICS codes (2007):

711410 Agents and managers for artists, athletes, entertainers, and other public figures [Professor Bieri, I have seen that you are considering this NAICS code, I am not 100% sure that we consider this one in the Artists variable since agents could include several artists. I think it might be better to consider both figures, i.e. agents and artists, as separate variables.]

711510 Independent artists, writers, and performers

```
> insheet using "E:\Research\Census-CBP\data\cbp10co.txt", clear
> keep if naics=="711410"|naics=="711510"
> gen FIPS = fipstate*1000+fipscty
> keep FIPS est
> collapse (sum) est , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if FIPS>2000&FIPS<3000
> drop if FIPS>15000&FIPS<16000
> sort FIPS
> replace est = 0 if missing(est)
> rename est artists
> label variable artists "Number of artists"
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> drop if _merge==1
> drop _merge
> gen artists_per1000 = artists/population*1000
> label variable artists_per1000 "Number of artists per 1000 people"
> order FIPS state county , first
> order population , last
> save "E:\Research\Census-CBP\Artist_2010_dhm.dta"
```

## 68. Movie theatres (per 1,000 people)

Data can be downloaded from the census website

([http://www.census.gov/econ/cbp/download/10\\_data/index.htm](http://www.census.gov/econ/cbp/download/10_data/index.htm)). Data is downloaded as .txt file and then imported into Stata. 'Movie theatres' has the following NAICS codes (2007):

512131 Motion picture theatres (except Drive-Ins)

512132 Drive-In motion picture theatres

```
> insheet using "E:\Research\Census-CBP\data\cbp10co.txt", clear
> keep if naics=="512131"|naics=="512132"
> gen FIPS = fipstate*1000+fipscty
> keep FIPS est
> collapse (sum) est , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if FIPS>2000&FIPS<3000
> drop if FIPS>15000&FIPS<16000
> sort FIPS
> replace est = 0 if missing(est)
> rename est movie
> label variable movie "Number of movie theatres"
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> drop if _merge==1
> drop _merge
> gen movie_per1000 = movie/population*1000
> label variable movie_per1000 "Number of movie theatres per 1000 people"
> order FIPS state county , first
> order population , last
> save "E:\Research\Census-CBP\Movies_2010_dhm.dta"
```

## 69. Bowling alleys (per 1,000 people)

Data can be downloaded from the census website

([http://www.census.gov/econ/cbp/download/10\\_data/index.htm](http://www.census.gov/econ/cbp/download/10_data/index.htm)). Data is downloaded as .txt file and then imported into Stata. 'Bowling alleys' has the following NAICS code (2007):

713950 Bowling centers

```
> insheet using "E:\Research\Census-CBP\data\cbp10co.txt", clear
> keep if naics=="713950"
> gen FIPS = fipstate*1000+fipscty
> keep FIPS est
> collapse (sum) est , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if FIPS>2000&FIPS<3000
> drop if FIPS>15000&FIPS<16000
> sort FIPS
> replace est = 0 if missing(est)
> rename est bowling
> label variable bowling "Number of bowling centers"
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> drop if _merge==1
> drop _merge
> gen bowling_per1000 = bowling/population*1000
> label variable bowling_per1000 "Number of bowling centers per 1000 people"
> order FIPS state county , first
> order population , last
> save "E:\Research\Census-CBP\Bowling_2010_dhm.dta"
```

## 70. Amusement, recreation establishments (per 1,000 people)

Data can be downloaded from the census website

([http://www.census.gov/econ/cbp/download/10\\_data/index.htm](http://www.census.gov/econ/cbp/download/10_data/index.htm)). Data is downloaded as .txt file and then imported into Stata. 'Amusement, recreation establishments' has the following NAICS code (2007):

713990 All other amusement and recreation industries

```
> insheet using "E:\Research\Census-CBP\data\cbp10co.txt", clear
> keep if naics=="713990"
> gen FIPS = fipstate*1000+fipscty
> keep FIPS est
> collapse (sum) est , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if FIPS>2000&FIPS<3000
> drop if FIPS>15000&FIPS<16000
> sort FIPS
> replace est = 0 if missing(est)
> rename est amusement
> label variable amusement "Number of amusement and recreation establishments"
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> drop if _merge==1
> drop _merge
> gen amusement_per1000 = amusement/population*1000
> label variable amusement_per1000 "Number of amusement and recreation establishments per 1000 people"
> order FIPS state county , first
> order population , last
> save "E:\Research\Census-CBP\Amusement_2010_dhm.dta"
```

## 71. Research I universities (Carnegie classification)

I have yet not found the definition of Research I universities. However, it seems that Research I universities are currently classified as 'Research universities (very high research activity) – RU/VH'. The classifications are described in the following website: <http://classifications.carnegiefoundation.org/descriptions/basic.php>. Then,



we can download the Classifications Data File from this website:

<http://classifications.carnegiefoundation.org/resources/>.

Once the excel file has been downloaded, we can determine the value used to identify RU/VH universities. These universities have the value of '15' in the variable ccbasic. Then we need to import this file into Stata.

```
> import excel "E:\Research\Carnegie\data\cc2010_classification_data_file_08.05.2013.xls", sheet("Data")
firstrow clear
> keep UNITID NAME CITY STABBR CCBASIC
> keep if CCBASIC == 15
```

Then since the dataset contains city information but does not contain county information, we need to merge this dataset with the county/place relationship file.

```
> merge m:m state placename using "E:\Research\CountyPlace\CountyPlace2010.dta"
> drop if missing(NAME)
> sort UNITID
```

Once we have merged these files, we need to review the merged file and verify that the county that was assigned is the correct one. There may be an error when a city or place falls on two or more counties. Review the assigned county when the variable 'afact' is less than 1. Then, we can verify the county by looking for the university official address online and then look for the county in which the university's zip code falls on by going to this website: <http://www.naco.org/counties/pages/citysearch.aspx>. Once we have edited the dataset we can run the following commands:

```
> edit
> gen ResearchI = 1
> keep FIPS ResearchI
> collapse (sum) ResearchI , by FIPS
> collapse (sum) ResearchI , by(FIPS)
> label variable ResearchI "Research-I universities"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> sort FIPS
> replace ResearchI = 0 if missing(ResearchI)
> save "E:\Research\Carnegie\ResearchI_2010.dta"
```

## 72. Golf courses and country clubs

Data can be downloaded from the census website

([http://www.census.gov/econ/cbp/download/10\\_data/index.htm](http://www.census.gov/econ/cbp/download/10_data/index.htm)). Data is downloaded as .txt file and then imported into Stata. 'Golf courses and country clubs' has the following NAICS code (2007):

713910 Golf Courses and Country Clubs

```
> insheet using "E:\Research\Census-CBP\data\cbp10co.txt", clear
> keep if naics=="713910"
> gen FIPS = fipstate*1000+fipscty
> keep FIPS est
> collapse (sum) est , by(FIPS)
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> drop statecode _merge
> drop if FIPS>2000&FIPS<3000
> drop if FIPS>15000&FIPS<16000
> sort FIPS
> replace est = 0 if missing(est)
> rename est golf
> label variable golf "Number of golf courses and country clubs"
> merge 1:1 FIPS using "E:\Research\Population\pop_county2010.dta"
> drop if _merge==1
> drop _merge
> gen golf_per1000 = golf/population*1000
```



```
> label variable golf_per1000 "Number of golf courses and country clubs per 1000 people"
> order FIPS state county , first
> order population , last
> save "E:\Research\Census-CBP\Golf_2010_dhm.dta"
```

### 73. Military areas (percentage of total land area)

The National Transportation Atlas Database (NTAD) 2010 from the Research and Innovative Technology Administration – Bureau of Transportation Statistics can be ordered from the following website:

<https://1bts.rita.dot.gov/pdc/user/products/src/products.xml?p=3194&c=-1>

However, this data is only available in DVD, so you will have to order it.

Military areas 2010 data is found in the National Transportation Atlas Database 2010. We need to load the shapefile into ArcMap and then join this shapefile with the county land shapefile. In order to do this we need to do the following:

- (1) Add the downloaded layer into ArcMap. Project it as GCS NAD83 (ArcToolbox \ Data Management Tools \ Projections and Transformations \ Feature \ Project).
- (2) Then, since the data is in a geographic coordinate system, it needs to be projected. Project it into USA Contiguous Albers Equal Area Conic. We are using this projected coordinate system because we are going to calculate areas; therefore we need to keep areas with minimal distortion (ArcToolbox \ Data Management Tools \ Projections and Transformations \ Feature \ Project).
- (3) Since this shapefile contains the military area, we need to intersect it with the county\_land shapefile in order to obtain the military area on land. Run the intersect tool (ArcToolbox \ Analysis Tools \ Overlay \ Intersect). Then select the following options:
  - a. Input features: Military area shapefile & county land shapefile.
  - b. Output feature class: Path and name of the new shapefile. This shapefile will have just the federal land.
- (4) Then, we need to open the attribute tables of the newly created shapefile.
- (5) Add a new field named: 'mil\_mi2' and then right click on it and select calculate geometry. Calculate the area in square miles.
- (6) Export the attribute table as a txt file.

Then we can import this table into Stata and use the information to calculate the federal land as percentage of the total land of the county. Run the following commands to import these txt files into Stata and format the information:

```
> insheet using "E:\Research\USDT-NTAD\data\NTAD2010\polygon\milbase\milbase_county.txt"
> gen FIPS = statefp10*1000+countyfp10
> collapse (sum) mil_mi2 , by(FIPS)
> rename mil_mi2 milarea_mi2
> label variable milarea_mi2 "Military area in square miles"
> merge 1:1 FIPS using "E:\Research\FIPS\FIPS.dta"
> sort FIPS
> drop _merge statecode
> replace milarea_mi2 = 0 if missing(milarea_mi2)
> merge 1:1 FIPS using "E:\Research\Census-TIGER\CountyLand_2010.dta"
> drop _merge
> gen milarea_pct = milarea_mi2/land_mi2*100
> label variable milarea_pct "Military area in percentage (over land area)"
> order FIPS state county , first
> drop land_mi2
> save "E:\Research\USDT-NTAD\MilitaryBase\MilitaryBase_2010.dta", replace
```

## **74. Housing stress (=1 if > 30% of hholds distressed)**

Data source is USDS-ERS. Website: <http://www.ers.usda.gov/>. To get an better idea of what the purpose of doing County Typology Codes is, go to <http://www.ers.usda.gov/Briefing/Rurality/Typology/Methods/>.

Data availability: 2004 (for 2000), 1989 (for 1990), 1986 (for 1980), and 1979 (for 1970)

the housing stress (2000)  
low-education (2000)  
low-employment (2000)  
persistent poverty (1970, 1980, 1990, and 2000)  
population loss (1980, 1990, and 2000)  
retirement destination (1990 and 2000)

All data are stored in <http://www.ers.usda.gov/Data/TypologyCodes/>

For 2004, all the housing stress, persistent poverty and retirement destination data have been collected.

For 1989, only persistent poverty and retirement destination have been collected.

For 2004 and 1989, variables are in different code systems which require for bridging.

2004 raw data was downloaded and saved in [C:\Data\USDS-ERS\2004\ house+poverty+retirement\\_2004.xls](#). STATA data of housing stress (together with persistent poverty and retirement destination which are from the same source) was saved as [C:\Data\USDS-ERS\2004\house+poverty+retirement\\_2004.dta](#).

Data in year 1989, 1983, and 1974 were collected and saved in [C:\Data\USDS-ERS\Historic](#).

## **75. Persistent poverty (=1 if > 20% of pop. in poverty)**

Data source is USDS-ERS. Website: <http://www.ers.usda.gov/>. To get an better idea of what the purpose of doing County Typology Codes is, go to <http://www.ers.usda.gov/Briefing/Rurality/Typology/Methods/>.

Data availability: 2004 (for 2000), 1989 (for 1990), 1986 (for 1980), and 1979 (for 1970)

the housing stress (2000)  
low-education (2000)  
low-employment (2000)  
persistent poverty (1970, 1980, 1990, and 2000)  
population loss (1980, 1990, and 2000)  
retirement destination (1990 and 2000)

All data are stored in <http://www.ers.usda.gov/Data/TypologyCodes/>

For 2004, all the housing stress, persistent poverty and retirement destination data have been collected.

For 1989, only persistent poverty and retirement destination have been collected.

For 2004 and 1989, variables are in different code systems which require for bridging.

2004 raw data was downloaded and saved in [C:\Data\USDS-ERS\2004\ house+poverty+retirement\\_2004.xls](#). STATA data of persistent poverty (together with housing stress and retirement destination which are from the same source) was saved as [C:\Data\USDS-ERS\2004\house+poverty+retirement\\_2004.dta](#).

Data in year 1989, 1983, and 1974 were collected and saved in <C:\Data\USDS-ERS\Historic>.

## **76. Retirement destination (=1 if growth retirees > 15%)**

Data source is USDS-ERS. Website: <http://www.ers.usda.gov/>. To get an better idea of what the purpose of doing County Typology Codes is, go to <http://www.ers.usda.gov/Briefing/Rurality/Typology/Methods/>.

Data availability: 2004 (for 2000), 1989 (for 1990), 1986 (for 1980), and 1979 (for 1970)

the housing stress (2000)

low-education (2000)

low-employment (2000)

persistent poverty (1970, 1980, 1990, and 2000)

population loss (1980, 1990, and 2000)

retirement destination (1990 and 2000)

All data are stored in <http://www.ers.usda.gov/Data/TypologyCodes/>

For 2004, all the housing stress, persistent poverty and retirement destination data have been collected.

For 1989, only persistent poverty and retirement destination have been collected.

For 2004 and 1989, variables are in different code systems which require for bridging.

2004 raw data was downloaded and saved in [C:\Data\USDS-ERS\2004\house+poverty+retirement\\_2004.xls](C:\Data\USDS-ERS\2004\house+poverty+retirement_2004.xls).

STATA data of retirement destination (together with persistent poverty and housing stress which are from the same source) was saved as [C:\Data\USDS-ERS\2004\house+poverty+retirement\\_2004.dta](C:\Data\USDS-ERS\2004\house+poverty+retirement_2004.dta).

Data in year 1989, 1983, and 1974 were collected and saved in <C:\Data\USDS-ERS\Historic>.

## **77. Distance (km) to the nearest urban center**

Data source and method follows PRAO-JIE09: PARTRIDGE, M. D., D. S. RICKMAN, K. ALI, AND M. R. OLFERT (2009): “Agglomeration Spillovers and Wage and Housing Cost Gradients Across the Urban Hierarchy,” *Journal of International Economics*, 78(2), 126–140. A special manual on how to calculate this data was created and saved as [C:\Data\Incremental\\_Distance\\_2010\Incremental Distance Manual\\_20120308.docx](C:\Data\Incremental_Distance_2010\Incremental Distance Manual_20120308.docx).

## **78. Incr. distance to a metropolitan area of any size**

Data source and method follows PRAO-JIE09: PARTRIDGE, M. D., D. S. RICKMAN, K. ALI, AND M. R. OLFERT (2009): “Agglomeration Spillovers and Wage and Housing Cost Gradients Across the Urban Hierarchy,” *Journal of International Economics*, 78(2), 126–140. A special manual on how to calculate this data was created and saved as [C:\Data\Incremental\\_Distance\\_2010\Incremental Distance Manual\\_20120308.docx](C:\Data\Incremental_Distance_2010\Incremental Distance Manual_20120308.docx).

## **79. Incr. distance to a metro area > 250,000**

Data source and method follows PRAO-JIE09: PARTRIDGE, M. D., D. S. RICKMAN, K. ALI, AND M. R. OLFERT (2009): “Agglomeration Spillovers and Wage and Housing Cost Gradients Across the Urban

Hierarchy," *Journal of International Economics*, 78(2), 126–140. A special manual on how to calculate this data was created and saved as [C:\Data\Incremental\\_Distance\\_2010\Incremental Distance Manual\\_20120308.docx](C:\Data\Incremental_Distance_2010\Incremental Distance Manual_20120308.docx).

## 80. Incr. distance to a metro area > 500,000

Data source and method follows PRAO-JIE09: PARTRIDGE, M. D., D. S. RICKMAN, K. ALI, AND M. R. OLFERT (2009): "Agglomeration Spillovers and Wage and Housing Cost Gradients Across the Urban Hierarchy," *Journal of International Economics*, 78(2), 126–140. A special manual on how to calculate this data was created and saved as [C:\Data\Incremental\\_Distance\\_2010\Incremental Distance Manual\\_20120308.docx](C:\Data\Incremental_Distance_2010\Incremental Distance Manual_20120308.docx).

## 81. Incr. distance to a metro area > 1.5 million

Data source and method follows PRAO-JIE09: PARTRIDGE, M. D., D. S. RICKMAN, K. ALI, AND M. R. OLFERT (2009): "Agglomeration Spillovers and Wage and Housing Cost Gradients Across the Urban Hierarchy," *Journal of International Economics*, 78(2), 126–140. A special manual on how to calculate this data was created and saved as [C:\Data\Incremental\\_Distance\\_2010\Incremental Distance Manual\\_20120308.docx](C:\Data\Incremental_Distance_2010\Incremental Distance Manual_20120308.docx).

---

**Data above are those in the LIST. However, David would like a couple of more data from the similar source. Here is information about the data beyond the LIST. Without comparable information, it is difficult to tell whether they pass the quality control test.**

## 82. Gambling

**NOTICE: 2010 CBP is scheduled for release in June 2012. NOTICE: CBP 2000 for 2000 data, 1990 for 1990 data, 1980 for 1980 data, but 1977 for 1970 data. .**

Gambling

7132 Gambling industries

The latest data is 2009, using NAICS 2007. Data is accessible from <http://www.census.gov/econ/cbp/download/index.htm>. Raw data is downloaded and stored as <C:\Data\CBP2009\cbp09co.txt>. Bowling alleys data was selected and saved as <C:\Data\CBP2009\Gambling2009.dta>.

## 83. Land Grant University

Land grant university is not a category in Carnegie classification system. Data are collected manually from [http://en.wikipedia.org/wiki/List\\_of\\_land-grant\\_universities](http://en.wikipedia.org/wiki/List_of_land-grant_universities). A spreadsheet was created and saved as [C:\Data\Land\\_Grant\\_College\Land\\_Grant\\_College.xls](C:\Data\Land_Grant_College\Land_Grant_College.xls). STATA data was saved as [C:\Data\Land\\_Grant\\_College\Land\\_Grant\\_Coll.dta](C:\Data\Land_Grant_College\Land_Grant_Coll.dta).

Notice that some universities have flagship campus and common wealth campuses. Flagship campus has been used for location (FIPS).

## PART 2: DATA SOURCE

### 1. NOAA-NCDC

### 2. ICPSR

### 3. NOAA-SEAD

### 4. ESRI

### 5. USDI-NPS

### 6. USGS-NA

### 7. EPA-TRI

Data stored by year in <http://www.epa.gov/tri/tridata/data/basic/index.html>.

TRI facility ID reference: [http://www.epa.gov/enviro/html/tris/column/tri\\_facility\\_id.html](http://www.epa.gov/enviro/html/tris/column/tri_facility_id.html)

Different facilities in EPA are assigned SIC and NAICS codes

I have also found a document titled *List of Treatment, Storage, and Disposal Facilities in the United States The Preliminary Biennial RCRA Hazardous Waste Report (Based on 1993 Data)* which exhibits the same format as Biennial Report, and thus I wonder whether the facilities listed in BR are Treatment, Storage, and Disposal Facilities. If they are, I will refer to BR in 2009, 2001, 1991 for data for 2010, 2000, and 1990 separately. An example of LIST OF REPORTED RCRA SITES IN THE UNITED STATES

<http://www.epa.gov/osw/inforesources/data/br09/list09.pdf>

How to access TSDF and LQGs (update on July 24th with the instruction of David):

- According to <http://www.epa.gov/waste/inforesources/data/brs99/brshelp.pdf>, data are stored in USOS1-BS123-FORM IC and comment in USOS5-BS5-FORM IC.
- Download data in flatfile format from ftp [ftp://ftp.epa.gov/rcrainfodata/br\\_1999/](ftp://ftp.epa.gov/rcrainfodata/br_1999/)
- Follow David's example and write the dictionary for it

## On-Site Waste Management Status RCRA T/D/R

Table: BS123\_FORM\_IC\_PART1

Data Element Name:

On-Site Waste Management Status RCRA T/D/R

Description:

Code indicating whether treatment, recycling, or disposal of hazardous wastes was conducted during the current year on-site in units requiring a RCRA permit.

Source:

Form IC, Section V, Part A

Format:

CHAR(1)

Allowed Values:

- 1 No, and the site does not have firm plans to develop an on-site RCRA permitted treatment, disposal, or recycling system.
- 2 No, but the site does have firm plans to develop an on-site RCRA permitted treatment, disposal, or recycling system.
- 3 Yes, hazardous waste was treated, disposed, or recycled, on-site during the current year in a unit subject to RCRA permitting requirements.

Stata/IC 11.2 - [Results]

File Edit Data Graphics Statistics User Window Help

Data Editor (Browse) - [Untitled]

File Edit Data Tools

Report[130291] 1991

|        | Generator1 | Generator2 | OnSite | Manage1 | OnSite2 | Storage | LocState | MailState | OnSite3 | Manage2 |  |
|--------|------------|------------|--------|---------|---------|---------|----------|-----------|---------|---------|--|
| 136660 | HQ         | 1          | 3      | HQ      | 5       | HQ      | MI       | MI        | 1       | US      |  |
| 136661 | HQ         | 4          | 3      | HQ      | 1       | HQ      | TX       | TX        |         |         |  |
| 136662 | HQ         | 1          | 3      | HQ      | 2       | HQ      | CA       | CA        | 3       | US      |  |
| 136663 | HQ         | 1          | 3      | HQ      | 8       | HQ      | IN       | IN        |         |         |  |
| 136664 | HQ         | 1          | 3      | HQ      | 1       | HQ      | UT       | UT        |         |         |  |
| 136665 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136666 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136667 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136668 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136669 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136670 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136671 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | IL        |         |         |  |
| 136672 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136673 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136674 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136675 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136676 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136677 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |
| 136678 | HQ         | 1          | H      | HQ      | H       | HQ      | OH       | OH        |         |         |  |

Vars: 71 Obs: 136,678 Filter: Off Mode: Browse CAP NUM

Reason5 Reason for Not... str1 %s  
Reason6 Reason for Not... str1 %s  
Reason7 Reason for Not... str1 %s  
SourceAct Source Reducti... str1 %s

E:\2011-2012 FALL\Research\Software\State\Guide from David

开始 10:51 PM

## 8. USDOE-INSC

## 9. EPA-AQS

## 10. EPA-NATA

## 11. COG

The 2012 COG is coming out soon. Here was the previous findings which might not apply to the new data.

|                                   |     |      |                              |   |                                       |
|-----------------------------------|-----|------|------------------------------|---|---------------------------------------|
| Local direct general expenditures | DDD | C301 | Total Expenditure            | N | Total expenditure                     |
|                                   | DDD | C302 | Total IG Expenditure         | N | Total intergovernmental expenditure   |
|                                   | DDD | C303 | Direct Expenditure           | N | Direct expenditure                    |
|                                   | E-- | C305 | Total Current Oper           | N | Total current operations              |
|                                   | DDD | C307 | Total Capital Outlays        | N | Total capital outlays                 |
|                                   | DDD | C311 | Tot Assist & Subsidies       | N | Total assistance and subsidies        |
|                                   | I-- | C312 | Total Interest on Debt       | N | Total interest on debt                |
|                                   | DDD | C313 | Total Insur Trust Ben        | N | Total insurance trust benefits        |
|                                   |     |      |                              |   |                                       |
|                                   | DDD | C304 | Total Current Expend         | N | Total current expenditure             |
|                                   |     |      |                              |   |                                       |
|                                   | Z00 | Z00  | Total Salaries & Wages (Z00) | N | Exhibit: Total salaries and wages     |
|                                   |     |      |                              |   |                                       |
|                                   | DDD | C315 | General Expenditure          | N | General expenditure                   |
|                                   |     |      |                              |   | Intergovernmental expenditure detail: |
|                                   | DDD | C316 | IG Exp-To State Govt         | N | To state government                   |
|                                   | DDD | C317 | IG Exp-To Local Govts        | N | To other local governments            |
|                                   | DDD | C324 | IG Exp-To Federal Govt       | N | To Federal Government                 |
|                                   | DDD | C325 | Direct General Expend        | N | Direct general expenditure            |

|  |     |       |                                   |   |                                     |
|--|-----|-------|-----------------------------------|---|-------------------------------------|
|  | DDD | C329  | General Capital Outlay            | N | General capital outlays             |
|  |     |       |                                   |   |                                     |
|  | DDD | C326  | General Current Expend            | N | General current expenditure         |
| Local exp. for hospitals and health  | DDD | C557  | Health & Hosp-Tot Exp             | N | Health and Hospitals, Total         |
|  |     |       |                                   |   |                                     |
|  | -32 | C562  | Health-Total Expend               | N | Health, Total                       |
|  |     |       |                                   |   |                                     |
|  | DDD | C577  | Total Hospital-Tot Exp            | N | Hospitals, Total                    |
|  | DDD | C8003 | Total Hospital-Other Than Capital | N | Other than capital outlays          |
| Local exp. on parks, rec. and nat. resources   | DDD | C580  | Total Hospital-Cap Out            | N | Capital outlays                     |
|  | DDD | C690  | Total Nat Res-Tot Exp             | N | Natural Resources, Total            |
|  | DDD | C8039 | Total Nat Res-Other Than Capital  | N | Other than capital outlays          |
|  | DDD | C694  | Total Nat Res-Cap Out             | N | Capital outlays                     |
|  |     |       |                                   |   |                                     |
| Teacher-pupil ratio (not found)  | -61 | C770  | Parks & Rec-Total Exp             | N | Parks and Recreation, Total         |
|  |     |       |                                   |   |                                     |
|  |     |       |                                   |   |                                     |
|  |     |       |                                   |   |                                     |
|  |     |       |                                   |   |                                     |
| Local expenditure per student (use C408 / total number of student which has not been found in COG) | DDD | C406  | Total Educ-Total Exp              | N | Education, Total                    |
|  | DDD | C8001 | Total Educ-Other Than Capital     | N | Other than capital outlays          |
|  | DDD | C410  | Total Educ-Cap Outlay             | N | Capital outlays                     |
| Federal expenditure  |     |       |                                   |   |                                     |
|  | DDD | C301  | Total Expenditure                 | N | Total expenditure                   |
|  | DDD | C302  | Total IG Expenditure              | N | Total intergovernmental expenditure |
|  | DDD | C303  | Direct Expenditure                | N | Direct expenditure                  |
|  | E-- | C305  | Total Current Oper                | N | Total current operations            |
|  | DDD | C307  | Total Capital Outlays             | N | Total capital outlays               |
|  | DDD | C311  | Tot Assist & Subsidies            | N | Total assistance and subsidies.     |



|     |      |                              |   |                                       |
|-----|------|------------------------------|---|---------------------------------------|
| I-- | C312 | Total Interest on Debt       | N | Total interest on debt                |
| DDD | C313 | Total Insur Trust Ben        | N | Total insurance trust benefits        |
|     |      |                              |   |                                       |
| DDD | C304 | Total Current Expend         | N | Total current expenditure             |
|     |      |                              |   |                                       |
| Z00 | Z00  | Total Salaries & Wages (Z00) | N | Exhibit: Total salaries and wages     |
|     |      |                              |   |                                       |
| DDD | C315 | General Expenditure          | N | General expenditure                   |
|     |      |                              |   | Intergovernmental expenditure detail: |
| DDD | C316 | IG Exp-To State Govt         | N | To state government                   |
| DDD | C317 | IG Exp-To Local Govts        | N | To other local governments            |
| DDD | C324 | IG Exp-To Federal Govt       | N | To Federal Government                 |
| DDD | C325 | Direct General Expend        | N | Direct general expenditure            |
| DDD | C329 | General Capital Outlay       | N | General capital outlays               |
|     |      |                              |   |                                       |
| DDD | C326 | General Current Expend       | N | General current expenditure           |

## 12. CBP

**Latest:** 2012 is coming

**Frequency:** Annual

**Starting from:** 1977 (see <http://www.census.gov/econ/cbp/historical.htm>, and <http://fisher.lib.virginia.edu/collections/stats/cbp/>)

Since CBP is based on NAICS code (previously used SIC code), and NAICS/SIC keep updating their versions, for data in different years, we need to “bridge” codes to get the information for the same industry. See [C:\Data\CBP\NAICS to SIC bridge sample.docx](#) to understand bridging problem with tax/exempting and unchangeable. For data 1980 and 1977, industries are categorized in 2-digit SIC. I may need 4-digit SIC for bridging (see [C:\Data\CBP\NAICS - SIC Codes Transfer\\_20110621.xlsx](#)). Another problem is that it seems I have to download state by state. The following table can help to understand NAICS code issue.

|      | Most Suitable Year | Standard   |
|------|--------------------|------------|
| 2010 | 2008               | NAICS 2007 |
| 2000 | 2000               | NAICS 1997 |

|      |      |          |
|------|------|----------|
| 1990 | 1990 | SIC 1987 |
| 1980 | 1980 | SIC 1972 |
| 1970 | 1977 | SIC 1972 |

### 13. Census

### 14. USDS-ERS

**Latest:** 2004

2004 is the most current data; besides that only 1989, 1986 and 1979 are available. Data for 2010s cannot be found.

For 2004, all the housing stress, persistent poverty and retirement destination data have been collected; For 1989, only persistent poverty and retirement destination have been collected. In addition, data in each of the three years are in different code systems.

### 15. CDC-NCHS

### 16. CCIHE

### 17. PRAO-JIE09

## PART 3: OTHER DATA-RELATED WORK

### 1. Compare LIST with D3's list

D3 list here refers to **XXXXX**. Comparison results are saved in [C:\Data\comparison\\_20110624.xls](#).

### 2. Overview of Existing Sustainability Indicators

## APPENDIX

### 1: Status track table

| Amenities                             | Units                                  | Data Source | 2010 | 2000 | 1990 | 1980 | 1970 |
|---------------------------------------|--|-------------|------|------|------|------|------|
| <b>GEOGRAPHY AND CLIMATE</b>          |  |             |      |      |      |      |      |
| Mean precipitation                    | inches p.a.                            | NOAA-NCDC   |      |      |      |      |      |
| Mean relative annual humidity         | %                                      | NOAA-NCDC   |      |      |      |      |      |
| Mean annual heating degree days (HDD) |  | NOAA-NCDC   |      |      |      |      |      |
| Mean annual cooling degree days (CDD) |  | NOAA-NCDC   |      |      |      |      |      |
| Mean wind speed                       | m.p.h.                                 | NOAA-NCDC   |      |      |      |      |      |
| Sunshine                              | % of possible                          | NOAA-NCDC   |      |      |      |      |      |
| Heavy fog                             | no. of days with visibility _ 0.25 mi. | NOAA-NCDC   |      |      |      |      |      |
| Percent water area                    |  | ICPSR       |      |      |      |      |      |
| Coast                                 | =1 if on coast                         | NOAA-SEAD   |      |      |      |      |      |
| Non-adjacent coastal watershed        | =1 if in watershed                     | NOAA-SEAD   |      |      |      |      |      |
| Mountain peaks above 1500 meters      |  | ESRI        |      |      |      |      |      |
| Rivers                                | miles per sq. mile                     | USDI-NPS    |      |      |      |      |      |
| Federal land                          | percentage of total land area          | USGS-NA     |      |      |      |      |      |
| Wilderness areas                      | percentage of total land area          | USGS-NA     |      |      |      |      |      |
| National Parks                        | percentage of total land area          | USGS-NA     |      |      |      |      |      |
| Distance to nearest National Park     | km                                     | USDI-NPS    |      |      |      |      |      |
| Distance to nearest State Park        | km                                     | USDI-NPS    |      |      |      |      |      |
| Scenic drives                         | total mileage                          | USGS-NA     |      |      |      |      |      |

|  |                          |            |  |  |  |  |  |
|--|--------------------------|------------|--|--|--|--|--|
| Average number of tornados per annum         |                          | USGS-NA    |  |  |  |  |  |
| Property damage from hazard events           | \$000s, per mi2          | USGS-NA    |  |  |  |  |  |
| Seismic hazard                               | index                    | USGS-NA    |  |  |  |  |  |
| Number of earthquakes                        |                          | USGS-NA    |  |  |  |  |  |
| Land cover diversity                         | index, range 0–255       | USGS-NA    |  |  |  |  |  |
| <b>ENVIRONMENTAL EXTERNALITIES</b>           |                          |            |  |  |  |  |  |
| NPDES effluent dischargers                   | PCS permits              | EPA-TRI    |  |  |  |  |  |
| Landfill waste                               | metric tons              | EPA-TRI    |  |  |  |  |  |
| Superfund sites                              |                          | EPA-TRI    |  |  |  |  |  |
| Treatment, storage and disposal facilities   |                          | EPA-TRI    |  |  |  |  |  |
| Large quantity generators of hazardous waste |                          | EPA-TRI    |  |  |  |  |  |
| Nuclear power plants                         |                          | USDOE-INSC |  |  |  |  |  |
| PM2.5  | µg per m <sup>3</sup>    | EPA-AQS    |  |  |  |  |  |
| PM10   | µg per m <sup>3</sup>    | EPA-AQS    |  |  |  |  |  |
| Ozone  | µg per m <sup>3</sup>    | EPA-AQS    |  |  |  |  |  |
| Sulphur dioxide                              | µg per m <sup>3</sup>    | EPA-AQS    |  |  |  |  |  |
| Carbon monoxide                              | µg per m <sup>3</sup>    | EPA-AQS    |  |  |  |  |  |
| Nitrogen dioxide                             | µg per m <sup>3</sup>    | EPA-AQS    |  |  |  |  |  |
| National Fire Plan treatment                 | percentage of total area | USGS-NA    |  |  |  |  |  |
| Cancer Risk                                  |                          | EPA-NATA   |  |  |  |  |  |
| Neurological risk                            |                          | EPA-NATA   |  |  |  |  |  |
| Respiratory risk                             |                          | EPA-NATA   |  |  |  |  |  |
| <b>LOCAL PUBLIC GOODS</b>                    |                          |            |  |  |  |  |  |
| Local direct general expenditures            | \$ per capita            | COG97      |  |  |  |  |  |
| Local exp. for hospitals and health          | \$ per capita            | COG97      |  |  |  |  |  |

|  |                               |             |  |  |  |  |  |
|--|-------------------------------|-------------|--|--|--|--|--|
| Local exp. on parks, rec. and nat. resources | \$ pc                         | COG97       |  |  |  |  |  |
| Museums and historical sites                 | per 1,000 people              | CBP         |  |  |  |  |  |
| Municipal parks                              | percentage of total land area | ESRI        |  |  |  |  |  |
| Campgrounds and camps                        |                               | CBP         |  |  |  |  |  |
| Zoos, botanical gardens and nature parks     |                               | CBP         |  |  |  |  |  |
| Crime rate                                   | per 100,000 persons           | ICPSR       |  |  |  |  |  |
| Teacher-pupil ratio                          |                               | COG97       |  |  |  |  |  |
| Local expenditure per student                | (\$, 1996-97 fiscal year      | COG97       |  |  |  |  |  |
| Private school to public school enrollment   | %                             | 2000 CENSUS |  |  |  |  |  |
| Child mortality                              | per 1000 births               | CDC-NCHS    |  |  |  |  |  |
| <b>INFRASTRUCTURE</b>                        |                               |             |  |  |  |  |  |
| Federal expenditure                          | \$ pc, non-wage, non-defense) | COG97       |  |  |  |  |  |
| Number of airports                           |                               | USGS-NA     |  |  |  |  |  |
| Number of ports                              |                               | USGS-NA     |  |  |  |  |  |
| Interstate highways                          | total mileage per mi2         | USGS-NA     |  |  |  |  |  |
| Urban arterial                               | total mileage per mi2         | USGS-NA     |  |  |  |  |  |
| Number of Amtrak stations                    |                               | USGS-NA     |  |  |  |  |  |
| Number of urban rail stops                   |                               | USGS-NA     |  |  |  |  |  |
| Railways                                     | total mileage per mi2         | USGS-NA     |  |  |  |  |  |
| <b>CULTURAL AND URBAN AMENITIES</b>          |                               |             |  |  |  |  |  |
| Number of restaurants and bars               | per 1,000 people              | CBP         |  |  |  |  |  |
| Theatres and musicals                        | per 1,000 people)             | CBP         |  |  |  |  |  |
| Artists                                      | per 1,000 people)             | CBP         |  |  |  |  |  |

|   |                                   |            |  |  |  |  |  |
|---|-----------------------------------|------------|--|--|--|--|--|
| Movie theatres                                    | per 1,000 people                  | CBP        |  |  |  |  |  |
| Bowling alleys                                    | per 1,000 people                  | CBP        |  |  |  |  |  |
| Amusement, recreation establishments              | per 1,000 people                  | CBP        |  |  |  |  |  |
| Research I universities (Carnegie classification) |                                   | CCIHE      |  |  |  |  |  |
| Suggested Landground University? (search)         |                                   |            |  |  |  |  |  |
| Golf courses and country clubs                    |                                   | CBP        |  |  |  |  |  |
| Military areas                                    | percentage of total land area     | USGS-NA    |  |  |  |  |  |
| Housing stress                                    | =1 if > 30% of hholds distressed) | USDS-ERS   |  |  |  |  |  |
| Persistent poverty                                | =1 if > 20% of pop. in poverty    | USDS-ERS   |  |  |  |  |  |
| Retirement destination                            | =1 if growth retirees > 15%       | USDS-ERS   |  |  |  |  |  |
| Distance to the nearest urban center              |                                   | PRAO-JIE09 |  |  |  |  |  |
| Incr. distance to a metropolitan area of any size |                                   | PRAO-JIE09 |  |  |  |  |  |
| Incr. distance to a metro area > 250,000          |                                   | PRAO-JIE09 |  |  |  |  |  |
| Incr. distance to a metro area > 500,000          |                                   | PRAO-JIE09 |  |  |  |  |  |
| Incr. distance to a metro area > 1.5 million      |                                   | PRAO-JIE09 |  |  |  |  |  |