DEMOGRAPHICS AND EMPLOYMENT IN THE UNITED STATES

Anuranjan

23 September 2016

In the United States, the government measures unemployment using the Current Population Survey (CPS), which collects demographic and employment information from a wide range of Americans each month.

CPS dataset represents people surveyed in the September 2013 CPS who actually completed a survey.
MetroAreaCodes dataset contains the mapping from codes to names of metropolitan areas.
CountryCodes dataset contains tThe mapping from codes to names of countries.

Reading CSV File

```
CPS=read.csv("CPSData.csv")
MetroAreaMap=read.csv("MetroAreaCodes.csv")
CountryMap=read.csv("CountryCodes.csv")
summary(CPS)
```

```
##  PeopleInHousehold       Region               State       MetroAreaCode
##  Min.   : 1.000    Midwest  :30684   California  :11570   Min.   :10420
##  1st Qu.: 2.000    Northeast:25939   Texas       : 7077   1st Qu.:21780
##  Median : 3.000    South    :41502   New York    : 5595   Median :34740
##  Mean   : 3.284    West     :33177   Florida     : 5149   Mean   :35075
##  3rd Qu.: 4.000                      Pennsylvania: 3930   3rd Qu.:41860
##  Max.   :15.000                      Illinois    : 3912   Max.   :79600
##                                      (Other)     :94069   NA's   :34238
##       Age                 Married          Sex
##  Min.   : 0.00   Divorced     :11151   Female:67481
##  1st Qu.:19.00   Married      :55509   Male  :63821
##  Median :39.00   Never Married:30772
##  Mean   :38.83   Separated    : 2027
##  3rd Qu.:57.00   Widowed      : 6505
##  Max.   :85.00   NA's         :25338
##
##                    Education                   Race
##  High school            :30906   American Indian :  1433
##  Bachelor's degree      :19443   Asian           :  6520
##  Some college, no degree:18863   Black           : 13913
##  No high school diploma :16095   Multiracial     :  2897
##  Associate degree       : 9913   Pacific Islander:   618
##  (Other)                :10744   White           :105921
##  NA's                   :25338
##     Hispanic      CountryOfBirthCode               Citizenship
##  Min.   :0.0000   Min.   : 57.00     Citizen, Native     :116639
##  1st Qu.:0.0000   1st Qu.: 57.00     Citizen, Naturalized:  7073
##  Median :0.0000   Median : 57.00     Non-Citizen         :  7590
##  Mean   :0.1393   Mean   : 82.68
##  3rd Qu.:0.0000   3rd Qu.: 57.00
##  Max.   :1.0000   Max.   :555.00
##
##            EmploymentStatus                                  Industry
##  Disabled          : 5712   Educational and health services   :15017
##  Employed          :61733   Trade                             : 8933
##  Not in Labor Force:15246   Professional and business services: 7519
##  Retired           :18619   Manufacturing                     : 6791
##  Unemployed        : 4203   Leisure and hospitality           : 6364
##  NA's              :25789   (Other)                           :21618
##                             NA's                              :65060
```

```
str(CPS)
```

```
## 'data.frame':    131302 obs. of  14 variables:
##  $ PeopleInHousehold : int  1 3 3 3 3 3 3 2 2 2 ...
##  $ Region            : Factor w/ 4 levels "Midwest","Northeast",..: 3 3 3 3
3 3 3 3 3 3 ...
##  $ State             : Factor w/ 51 levels "Alabama","Alaska",..: 1 1 1 1 1
1 1 1 1 1 ...
##  $ MetroAreaCode     : int  26620 13820 13820 13820 26620 26620 26620 33660
33660 26620 ...
##  $ Age               : int  85 21 37 18 52 24 26 71 43 52 ...
##  $ Married           : Factor w/ 5 levels "Divorced","Married",..: 5 3 3 3
5 3 3 1 1 3 ...
##  $ Sex               : Factor w/ 2 levels "Female","Male": 1 2 1 2 1 2 2 1
2 2 ...
##  $ Education         : Factor w/ 8 levels "Associate degree",..: 1 4 4 6 1
2 4 4 4 2 ...
##  $ Race              : Factor w/ 6 levels "American Indian",..: 6 3 3 3 6 6
6 6 6 6 ...
##  $ Hispanic          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CountryOfBirthCode: int  57 57 57 57 57 57 57 57 57 57 ...
##  $ Citizenship       : Factor w/ 3 levels "Citizen, Native",..: 1 1 1 1 1 1
1 1 1 1 ...
##  $ EmploymentStatus  : Factor w/ 5 levels "Disabled","Employed",..: 4 5 1 3
2 2 2 2 3 2 ...
##  $ Industry          : Factor w/ 14 levels "Agriculture, forestry, fishing,
and hunting",..: NA 11 NA NA 11 4 14 4 NA 12 ...
```

```
summary(MetroAreaMap)
```

```
##       Code                                    MetroArea
##  Min.   :  460   Akron, OH                          :  1
##  1st Qu.:19800   Albany-Schenectady-Troy, NY        :  1
##  Median :30780   Albany, GA                         :  1
##  Mean   :31961   Albuquerque, NM                    :  1
##  3rd Qu.:41460   Allentown-Bethlehem-Easton, PA-NJ:  1
##  Max.   :79600   Altoona, PA                        :  1
##                  (Other)                            :265
```

```
str(MetroAreaMap)
```

```
## 'data.frame':    271 obs. of  2 variables:
##  $ Code     : int  460 3000 3160 3610 3720 6450 10420 10500 10580 10740 ...
##  $ MetroArea: Factor w/ 271 levels "Akron, OH","Albany-Schenectady-Troy, N
Y",..: 12 92 97 117 122 195 1 3 2 4 ...
```

```
summary(CountryMap)
```

```
##       Code                            Country
##  Min.   : 57.0   Afghanistan             :  1
##  1st Qu.:152.0   Africa, not specified   :  1
##  Median :235.0   Albania                 :  1
##  Mean   :262.8   Algeria                 :  1
##  3rd Qu.:362.0   Americas, not specified:  1
##  Max.   :555.0   Antigua and Barbuda     :  1
##                  (Other)                 :143
```

```
str(CountryMap)
```

```
## 'data.frame':    149 obs. of  2 variables:
##  $ Code   : int  57 66 73 78 96 100 102 103 104 105 ...
##  $ Country: Factor w/ 149 levels "Afghanistan",..: 139 57 105 135 97 3 11 1
8 24 37 ...
```

Some Max and Min parameters.

Among the interviewees with a value reported for the Industry variable, what is the most common industry of employment?

```
sort(table(CPS$Industry))[length(sort(table(CPS$Industry)))]
```

```
## Educational and health services
##                           15017
```

Which state has the fewest and largest interviewees?

```
sort(table(CPS$State))[1]
```

```
## New Mexico
##       1102
```

```
sort(table(CPS$State))[length(sort(table(CPS$State)))]
```

```
## California
##      11570
```

Races where there are at least 250 interviewees in the CPS dataset of Hispanic ethnicity.

```
CPS_Hispanic=subset(CPS,CPS$Hispanic != 0)
CPS_Race_atleast250_Hispanic=table(CPS_Hispanic$Race)
CPS_Race_atleast250_Hispanic[CPS_Race_atleast250_Hispanic > 249]
```

```
##
## American Indian           Black     Multiracial           White
##             304             621             448           16731
```

Which variables have at least one interviewee with a missing (NA) value i.e. which columns have atleast one NA value.

```
# or we can use names(which(colSums(is.na(CPS))>0))
colnames(CPS)[colSums(is.na(CPS)) > 0]
```

```
## [1] "MetroAreaCode"    "Married"          "Education"
## [4] "EmploymentStatus" "Industry"
```

Which region of the United States has the largest proportion of interviewees living in a non-metropolitan area? (aka they have a missing MetroAreaCode value)

```
table(CPS$Region, is.na(CPS$MetroAreaCode))
```

```
##
##             FALSE  TRUE
##   Midwest   20010 10674
##   Northeast 20330  5609
##   South     31631  9871
##   West      25093  8084
```

Midwest has the most TRUE values i.e. missing MetroAreaCode values. But calculation proportion will involve manual work.

The **mean()** function, which takes the average of the values passed to it, will treat TRUE as 1 and FALSE as 0, meaning it returns the proportion of values that are true. For instance, mean(c(TRUE, FALSE, TRUE, TRUE)) returns 0.75. Knowing this, we can use tapply() with the mean function to answer proportion type questions. e.g. Which state has a proportion of interviewees living in a non-metropolitan area closest to 30%?

```
tapply(is.na(CPS$MetroAreaCode), CPS$State, mean)
```

```
##              Alabama               Alaska              Arizona
##           0.25872093           1.00000000           0.13154450
##             Arkansas           California             Colorado
##           0.49049965           0.02048401           0.12991453
##          Connecticut             Delaware District of Columbia
##           0.08568406           0.23396567           0.00000000
##              Florida              Georgia               Hawaii
##           0.03923092           0.19843249           0.24916627
##                Idaho             Illinois              Indiana
##           0.49868248           0.11221881           0.29141717
##                 Iowa               Kansas             Kentucky
##           0.48694620           0.36227390           0.50678979
##            Louisiana                Maine             Maryland
##           0.16137931           0.59832081           0.06937500
##        Massachusetts             Michigan            Minnesota
##           0.06492199           0.17825661           0.31506849
##         Mississippi             Missouri              Montana
##           0.69430894           0.32867133           0.83607908
##             Nebraska               Nevada        New Hampshire
##           0.58132376           0.13308190           0.56874530
##           New Jersey           New Mexico             New York
##           0.00000000           0.24500907           0.08060769
##       North Carolina         North Dakota                 Ohio
##           0.37304315           0.73738602           0.25122349
##             Oklahoma               Oregon         Pennsylvania
##           0.32764281           0.21821925           0.17430025
##         Rhode Island       South Carolina         South Dakota
##           0.00000000           0.31302774           0.70250000
##            Tennessee                Texas                 Utah
##           0.35594170           0.14370496           0.21009772
##              Vermont             Virginia           Washington
##           0.65238095           0.19844226           0.18131868
##        West Virginia            Wisconsin              Wyoming
##           0.75585522           0.29932986           1.00000000
```

It will be easier to answer this question if the proportions are sorted, which can be accomplished with:

```
sort(tapply(is.na(CPS$MetroAreaCode), CPS$State, mean))
```

```
## District of Columbia          New Jersey         Rhode Island
##           0.00000000          0.00000000           0.00000000
##           California             Florida        Massachusetts
##           0.02048401          0.03923092           0.06492199
##             Maryland            New York          Connecticut
##           0.06937500          0.08060769           0.08568406
##             Illinois            Colorado              Arizona
##           0.11221881          0.12991453           0.13154450
##               Nevada               Texas           Louisiana
##           0.13308190          0.14370496           0.16137931
##         Pennsylvania            Michigan           Washington
##           0.17430025          0.17825661           0.18131868
##              Georgia            Virginia                 Utah
##           0.19843249          0.19844226           0.21009772
##               Oregon            Delaware           New Mexico
##           0.21821925          0.23396567           0.24500907
##               Hawaii                Ohio              Alabama
##           0.24916627          0.25122349           0.25872093
##              Indiana           Wisconsin       South Carolina
##           0.29141717          0.29932986           0.31302774
##            Minnesota            Oklahoma             Missouri
##           0.31506849          0.32764281           0.32867133
##            Tennessee              Kansas       North Carolina
##           0.35594170          0.36227390           0.37304315
##                 Iowa            Arkansas                Idaho
##           0.48694620          0.49049965           0.49868248
##             Kentucky       New Hampshire             Nebraska
##           0.50678979          0.56874530           0.58132376
##                Maine             Vermont          Mississippi
##           0.59832081          0.65238095           0.69430894
##         South Dakota         North Dakota        West Virginia
##           0.70250000          0.73738602           0.75585522
##              Montana              Alaska              Wyoming
##           0.83607908          1.00000000           1.00000000
```

we can see that Wisconsin is the state closest to having 30% of its interviewees from a non-metropolitan area

Merging two DataFrames using merge().

The first two arguments determine the data frames to be merged (they are called “x” and “y”, respectively, in the subsequent parameters to the merge function). by.x=“MetroAreaCode” means we’re matching on the MetroAreaCode variable from the “x” data frame (CPS), while by.y=“Code” means we’re matching on the Code variable from the “y” data frame (MetroAreaMap). Finally, all.x=TRUE means we want to keep all rows from the “x” data frame (CPS), even if some of the rows’ MetroAreaCode doesn’t match any codes in MetroAreaMap (for those familiar with database terminology, this parameter makes the operation a **left outer join** instead of an inner join).

```
CPS = merge(CPS, MetroAreaMap, by.x="MetroAreaCode", by.y="Code", all.x=TRUE)
```

Which metropolitan area has the highest proportion of interviewees of Hispanic ethnicity?

```
temp=sort(tapply(CPS$Hispanic,CPS$MetroArea,mean))
temp[length(temp)]
```

```
## Laredo, TX
##  0.9662921
```

Determine the number of metropolitan areas in the United States from which at least 20% of interviewees are Asian.

```
temp=sort(tapply(CPS$Race=="Asian",CPS$MetroArea,mean))
length(temp[temp>=0.20])
```

```
## [1] 4
```

Determine which metropolitan area has the smallest proportion of interviewees who have received no high school diploma.(Hint:ignore NA values as none of the interviewees aged 14 and younger have an education value reported, so the mean value is reported as NA for each metropolitan area.)

```
temp=sort(tapply(CPS$Education == "No high school diploma", CPS$MetroArea, mea
n,na.rm=TRUE))
temp[1]
```

```
## Iowa City, IA
##    0.02912621
```

Merging Country Code

```
CPS = merge(CPS, CountryMap, by.x="CountryOfBirthCode", by.y="Code", all.x=TRU
E)
```

How many interviewees have a missing value for the new country of birth variable?

```
temp=summary(CPS$Country)
temp["NA's"]
```

```
## NA's
##  176
```

What proportion of the interviewees from the “New York-Northern New Jersey-Long Island, NY-NJ-PA” metropolitan area have a country of birth that is not the United States

```
temp=table(CPS$MetroArea == "New York-Northern New Jersey-Long Island, NY-NJ-P
A", CPS$Country != "United States")
temp
```

```
##
##          FALSE  TRUE
##   FALSE 78757 12744
##   TRUE   3736  1668
```

Here, first condition come in rows and second form columns. Next, (FALSE and TRUE come in order). So to get the answer of the question, we can use

```
temp[2,2]/(temp[2,1]+temp[2,2])
```

```
## [1] 0.3086603
```

Which metropolitan area has the largest number (note – not proportion) of interviewees with a country of birth in India? To obtain the number of TRUE values in a vector of TRUE/FALSE values, you can use the sum() function. For instance, sum(c(TRUE, FALSE, TRUE, TRUE)) is 3.

```
temp=sort(tapply(CPS$Country == "India", CPS$MetroArea, sum, na.rm=TRUE))
temp[length(temp)]
```

```
## New York-Northern New Jersey-Long Island, NY-NJ-PA
##                                                 96
```