Working with National Crime Victimization Survey Data

Greg Ridgeway (gridge@upenn.edu)
Ruth Moyer (moyruth@upenn.edu)

October 07, 2018

Introduction

Through our work with the UCR, we've already discussed reported crime. Nonetheless, not all crimes are reported to the police. Also, sometimes the UCR doesn't provide us with specific information about a victim-involved crime incident such as whether the victim knew the offenders or the location of the crime incident.

Each year, the U.S. Census Bureau conducts the National Crime Victimization Survey (NCVS), which is a valuable source of self-reported victimization data. The Census Bureau interviews a sample of people about the number and characteristics of crime victimizations they experienced during the prior 6 months. In 2015, for example, they collected data from 95,760 households and 163,880 persons.

The NCVS contains valuable information about nonfatal personal crimes such as rape or robbery as well as property crimes such as burglary. Additional information about the NCVS can be found at the BJS website. To give a sense of the type of data that the NCVS contains, refer to the Official 2012-2013 BJS Crime Victimization report.

Acquiring the NCVS data

The University of Michigan consolidates the NCVS data into a format that is easily accessible in R. We will be using 2012 and 2013 NCVS data.

First, we will download the NCVS 2012 data, ICPSR 34650. We will need to download the following files, DS1, DS2, DS3, DS4, and DS5 in R format. Also, download DS0, the Codebook (which is in PDF format). We will refer to the codebook frequently. As for the DS1, DS2, DS3, DS4, and DS5 files, we are interested in the .rda files.

Next, downoad the NCVS 2013 data, ICPSR 35164. Same drill as above - retrieve DS1, DS2, DS3, DS4, and DS5 in R format.

All told you should have ten .rda files, and one PDF codebook. The codebook is extremely important for understanding what the variable names stand for, and you should become familiar with it as soon as you can. For now, we won't be using the DS5 files that much. Also, the file names are admittedly a bit unwieldy with all the numbers so it might be a good idea to change the names to something that will help you quickly distinguish among all the files. We've created subfolders called NCVS2012 and NCVS2013 that contains the files extracted from the data download. Here are the files we have in our NCVS2012 and NCVS2013 subfolders.

```r
list.files("NCVS2012/",recursive = TRUE)
```

```
 [1] "34650-Codebook.pdf"                  "34650-descriptioncitation.pdf"
 [3] "34650-manifest.txt"                  "34650-related_literature.txt"
 [5] "DS0001/34650-0001-Data.rda"          "DS0002/34650-0002-Data.rda"
 [7] "DS0003/34650-0003-Data.rda"          "DS0004/34650-0004-Data.rda"
 [9] "DS0005/34650-0005-Data.rda"          "factor_to_numeric_icpsr.R"
[11] "series-95-related_literature.txt" "TermsOfUse.html"
```

```r
list.files("NCVS2013/",recursive = TRUE)
```

```
 [1] "35164-Codebook.pdf"                  "35164-descriptioncitation.pdf"
 [3] "35164-manifest.txt"                  "35164-related_literature.txt"
 [5] "DS0001/35164-0001-Data.rda"          "DS0002/35164-0002-Data.rda"
 [7] "DS0003/35164-0003-Data.rda"          "DS0004/35164-0004-Data.rda"
 [9] "DS0005/35164-0005-Data.rda"          "factor_to_numeric_icpsr.R"
[11] "series-95-related_literature.txt" "TermsOfUse.html"
```

Let's see what's in these .rda files. The DS1s for both 2012 and 2013 are the address record-type files. First, 2012:

```r
load("NCVS2012/DS0001/34650-0001-Data.rda")
ls()
head(da34650.0001)
```

```
[1] "da34650.0001"
              V1001  YEARQ                      IDHH V1002
1 (1) Address record 2012.1 2501017260961929294229224 27296
2 (1) Address record 2012.1 2501051210759582293728435 24034
3 (1) Address record 2012.1 2501286218428920608853213 26233
4 (1) Address record 2012.1 2501382697440982298228224 27298
5 (1) Address record 2012.1 2501533299154388298804435 24033
6 (1) Address record 2012.1 2501586708146353299320324 27299
                 V1003 V1004                V1005 V1006 V1008 V1009
1 (121) 2012, 1st quarter    25 01017260961929294229     2    24  2012
2 (121) 2012, 1st quarter    25 01051210759582293728     4    35  2012
3 (121) 2012, 1st quarter    25 01286218428920608853     2    13  2012
4 (121) 2012, 1st quarter    25 01382697440982298228     2    24  2012
5 (121) 2012, 1st quarter    25 01533299154388298804     4    35  2012
6 (121) 2012, 1st quarter    25 01586708146353299320     3    24  2012
    V1010
1 6172013
2 6172013
3 6172013
4 6172013
5 6172013
6 6172013
```

As you can see, the DS1 for 2012 contains a unique identifer for each interviewed household. Let's load the address record-type file for 2013.

```r
load("NCVS2013/DS0001/35164-0001-Data.rda")
```

Let's give these address record-type files for 2012 and 2013 more useful names.

```r
dataAddr12 <- da34650.0001
dataAddr13 <- da35164.0001
```

By contrast, DS2 contains household information. Let's load the household data and give them more useful names.

```r
load("NCVS2012/DS0002/34650-0002-Data.rda")
load("NCVS2013/DS0002/35164-0002-Data.rda")

dataHH12 <- da34650.0002
dataHH13 <- da35164.0002
```

The DS3 files contain person specific information whereas the DS4 files provide incident information. Let's load them and give them useful names.

```r
load("NCVS2012/DS0003/34650-0003-Data.rda")
load("NCVS2013/DS0003/35164-0003-Data.rda")
dataPers12 <- da34650.0003
dataPers13 <- da35164.0003

load("NCVS2012/DS0004/34650-0004-Data.rda")
load("NCVS2013/DS0004/35164-0004-Data.rda")
dataInc12 <- da34650.0004
dataInc13 <- da35164.0004
```

Now that we've loaded and renamed all the files we'll need, we can remove objects from our working environment that we no longer need. We can use `rm()` to accomplish this:

```r
rm(da34650.0001,da34650.0002,da34650.0003,da34650.0004,
   da35164.0001,da35164.0002,da35164.0003,da35164.0004)
```

Let's examine in a bit more detail the first three rows of the person file. The dataset contains 240 columns so we will just show the first 40 columns here. Note IDHH (household ID), IDPER (person ID), and the relationship between the first two rows. Also, note that V3077 (Variable #3077) refers to who responded to the survey.

```r
dataPers12[1:3, 1:40]
```

```
              V3001  YEARQ                     IDHH
1 (3) Person record 2012.1 25010172609619292942292 24
2 (3) Person record 2012.1 25010172609619292942292 24
3 (3) Person record 2012.1 25010512107595822937284 35
                       IDPER V3002                   V3003 V3004
1 25010172609619292942292240 1 27296 (121) 2012, 1st quarter    25
2 25010172609619292942292240 2 27296 (121) 2012, 1st quarter    25
3 25010512107595822937284350 1 24034 (121) 2012, 1st quarter    25
                 V3005 V3006 V3008 V3009 V3010              V3011
1 01017260961929294229     2    24     1     1 (2) Telephone/self
```

```
2 01017260961929294229     2    24     2     2 (2) Telephone/self
3 01051210759582293728     4    35     1     1 (2) Telephone/self
                  V3012 V3013 V3014             V3015             V3016
1 (11) Reference person    22    22       (1) Married       (1) Married
2            (02) Wife    18    18       (1) Married       (1) Married
3 (11) Reference person    28    28 (5) Never married (6) Not inter last
       V3017      V3018   V3019                        V3020
1   (1) Male   (1) Male (1) Yes         (28) High school grad
2 (2) Female (2) Female  (2) No         (28) High school grad
3   (1) Male   (1) Male  (2) No (40) Some college(no degree)
           V3023A   V3024         V3025 V3026 V3027 V3031 V3032 V3033
1 (02) Black only  (2) No (02) February    27  2012    NA     9    NA
2 (01) White only (1) Yes (02) February     2  2012     9    NA     3
3 (01) White only  (2) No    (03) March    11  2012     5    NA     3
    V3034 V3035   V3036 V3037  V3038 V3039  V3040 V3041   V3042 V3043
1  (2) No    NA    <NA>    NA   <NA>    NA (2) No    NA  (2) No    NA
2  (2) No    NA  (2) No    NA (2) No    NA (2) No    NA  (2) No    NA
3 (1) Yes     1 (1) Yes     1 (2) No    NA (2) No    NA (1) Yes     2
```

Let's examine the corresponding household information. This dataset also has a lot of features so we will just show here the first 53 of 280 columns.

```r
subset(dataHH12, IDHH=="2501017260961929294229224")[,1:53]
```

```
                 V2001  YEARQ                      IDHH V2002
1 (2) Household record 2012.1 2501017260961929294229224 27296
                    V2003 V2004                V2005 V2006 V2008 V2009
1 (121) 2012, 1st quarter    25 01017260961929294229     2    24     0
                  V2010         V2011 V2012         V2013
1 (1) Unit in smpl/prev (1) Same hhld     2 (998) Residue
                V2014                V2015     V2016     V2017 V2018
1 (2) Rented for cash (2) Rented for cash (1) Urban (1) Urban  <NA>
           V2019               V2020               V2021          V2022
1 (7) Item blank (01) House/apt/flat (01) House/apt/flat (1) Phone/unit
    V2023     V2024   V2025  V2025A  V2025B             V2026 V2027 V2028
1 (1) Yes (04) Four (1) Yes (1) Yes (1) Yes (07) 17,500-19,999  <NA>    NA
  V2029                 V2030 V2031     V2032 V2033       V2034
1    NA (300) Interviewed hhld  <NA> (02) Wife    18 (1) Married
        V2035      V2036  V2037                 V2038          V2040A
1 (1) Married (2) Female (2) No (28) High school grad (01) White only
    V2041 V2042       V2043       V2044    V2045   V2046
1 (1) Yes    22 (1) Married (1) Married (1) Male (1) Yes
                  V2047           V2049A  V2050 V2051 V2052
1 (28) High school grad (02) Black only (2) No    NA    NA
```

And the corresponding incident file (just the first 43 of 950 columns):

```r
dataInc12[1:3, 1:43]
```

```
                V4001  YEARQ                      IDHH
```

```
1 (4) Incident record 2012.1 2501051210759582293728435
2 (4) Incident record 2012.1 2501051210759582293728435
3 (4) Incident record 2012.1 2501051210759582293728435
                        IDPER V4002                   V4003 V4004
1 2501051210759582293728435 01 24034 (121) 2012, 1st quarter    25
2 2501051210759582293728435 01 24034 (121) 2012, 1st quarter    25
3 2501051210759582293728435 01 24034 (121) 2012, 1st quarter    25
                 V4005 V4006 V4008 V4009 V4010                V4011
1 0105121075958229372 8     4    35     1     1 (36) 36:Indiv scrn quest
2 0105121075958229372 8     4    35     1     1  (37) 37:Hhld scrn quest
3 0105121075958229372 8     4    35     1     1 (41) 41:Indiv scrn quest
  V4012              V4013         V4014 V4015 V4016             V4017
1     1 (2) Bef mov this add (09) September  2011     1 (1) 1-5 incidents
2     1 (2) Bef mov this add (09) September  2011     1 (1) 1-5 incidents
3     1 (2) Bef mov this add (09) September  2011     2 (1) 1-5 incidents
  V4018 V4019            V4021B             V4022  V4023 V4023B
1  <NA>  <NA> (01) Aft 6am-12am (4) Diff city etc (2) No (2) No
2  <NA>  <NA> (01) Aft 6am-12am (4) Diff city etc (2) No (2) No
3  <NA>  <NA> (06) Aft 9pm-12pm (4) Diff city etc (2) No (2) No
                 V4024  V4025   V4026 V4027   V4028              V4029
1  (02) R/hme-det bldg (2) No (1) Yes  <NA> (1) Yes (1) At least 1 entry
2 (01) R/hme-own dwell (2) No (1) Yes  <NA>  (2) No               <NA>
3   (12) Comm-rest/bar   <NA>    <NA>  <NA>    <NA>               <NA>
   V4030  V4031  V4032  V4033  V4034  V4035   V4036  V4037  V4038
1 (0) No (0) No (0) No (0) No (0) No (0) No (1) Yes (0) No (0) No
2   <NA>   <NA>   <NA>   <NA>   <NA>   <NA>    <NA>   <NA>   <NA>
3   <NA>   <NA>   <NA>   <NA>   <NA>   <NA>    <NA>   <NA>   <NA>
                V4039              V4040 V4041A
1 (0) No out of range               <NA>   <NA>
2                <NA> (04) Unlk door/win   <NA>
3                <NA>               <NA>   <NA>
```

Let's look at the month and year of crime incident variables

```
with(dataInc12, table(V4014,V4015))
with(dataInc13, table(V4014,V4015))
```

```
                V4015
V4014            2011 2012
  (01) January      0  728
  (02) February     0  658
  (03) March        0  705
  (04) April        0  751
  (05) May          0  768
  (06) June         0  825
  (07) July       159  670
  (08) August     296  560
  (09) September  366  426
  (10) October    492  298
```

```
  (11) November   608  139
  (12) December   766    0
  (98) Residue      0    0
               V4015
V4014            2012 2013
  (1) January      0  566
  (2) February     0  580
  (3) March        0  615
  (4) April        0  526
  (5) May          0  688
  (6) June         0  649
  (7) July       144  580
  (8) August     245  474
  (9) September  306  306
  (10) October   440  238
  (11) November  557  116
  (12) December  697    0
  (98) Residue     0    0
```

Creating a dataframe and weights with NCVS incident data

Next, we can create a 2012 incident dataframe. Importantly, the 2012 data contain incidents that occurred in 2012 as well as 2011 but were all self-reported to the Census Bureau in 2012. Likewise, the 2013 data contain incidents that occurred in 2012 as well as 2013. If we wanted to analyze crime that occurred in only 2012, we'd subset the data to include only 2012. We will combine the 2012 and 2013 incident dataframes and then subset this new dataframe so that we exclude 2011 and 2013. As we can see in the Codebook PDF, the variable V4015 refers to the year of occurrence. (Helpful hint: the numbering of the variables correlate to the numbering of the dataframe. The incident-level file is DS4. Many of the variables in DS4 are V4XXX.)

`rbind` binds rows. This is good for when the columns in two datasets are exactly the same.

```r
dataInc <- rbind(dataInc12,dataInc13)
table(dataInc$V4015) # year crime occured
dataInc <- subset(dataInc, V4015==2012)
```

```
2011 2012 2013
2687 8917 5338
```

We will also want to exclude crime that happens outside the United States or crimes for which we do not know the location (NA). According to the Codebook, V4022 refers to location.

```r
dataInc <- subset(dataInc, (V4022!="(1) Outside U.S.") | is.na(V4022))
```

A lot of crimes happen in a series. The BJS convention is to include up to 10 occurrences in a series crime.

```
i <- with(dataInc, which((V4019=="(2) No (is series)") & (V4016>=11) & (V4016<=996)))
dataInc$V4016[i] <- 10
dataInc$V4016[dataInc$V4016>=997] <- NA
```

Also, BJS analyses of NCVS data generally use weights because NCVS is survey data. We want to weight the survey data so that they are representative of the wider U.S. population! There are three NCVS weight categories: household, personal, and incident.

For more information about NCVS weights, consult the section on Weighting Information found at this ICPSR resource guide to the NCVS: (https://www.icpsr.umich.edu/icpsrweb/NACJD/NCVS/accuracy.jsp).

To that extent, let's update the weight for series crimes and create a "date year" weight.

```
i <- which(dataInc$V4019=="(2) No (is series)")
dataInc$WGTVICDY <- dataInc$WGTVICCY
dataInc$WGTVICDY[i] <- with(dataInc, WGTVICDY[i] * V4016[i])
```

We can also tabulate total weight by crime type to estimate the count of a crime. As the Codebook instructs, `V4529` is the variable for crime type.

```
aggregate(WGTVICDY~V4529, data=dataInc, sum)
```

```
                     V4529    WGTVICDY
1        (01) Completed rape   74309.666
2        (02) Attempted rape   59501.772
3     (03) Sex aslt w s aslt   41212.611
4     (04) Sex aslt w m aslt    6515.781
5      (05) Rob w inj s aslt   79343.272
6      (06) Rob w inj m aslt   77564.887
7          (07) Rob wo injury  176027.246
8      (08) At rob inj s asl   28969.151
9      (09) At rob inj m asl   26869.716
10        (10) At rob w aslt  148857.011
11     (11) Ag aslt w injury  385348.494
12     (12) At ag aslt w wea  271055.951
13      (13) Thr aslt w weap  421411.004
14      (14) Simp aslt w inj  954981.736
15      (15) Sex aslt wo inj   32580.327
16     (16) Unw sex wo force   15992.059
17 (17) Asl wo weap, wo inj 2005635.943
18      (18) Verbal thr rape   39745.499
19     (19) Ver thr sex aslt   15369.782
20      (20) Verbal thr aslt 2019545.074
21      (21) Purse snatching   15990.538
22      (22) At purse snatch    7272.660
23       (23) Pocket picking  126418.096
24      (31) Burg, force ent 1215286.994
25     (32) Burg, ent wo for 1758044.551
26      (33) Att force entry  711352.327
```

```
27      (40) Motor veh theft  480278.161
28     (41) At mtr veh theft  165996.837
29          (54) Theft < $10 1115139.162
30        (55) Theft $10-$49 2899929.059
31       (56) Theft $50-$249 4918627.396
32          (57) Theft $250+ 3790419.581
33       (58) Theft value NA 1369499.977
34      (59) Attempted theft  686151.735
35        (1) Completed rape   54822.944
36        (2) Attempted rape    1640.455
37     (3) Sex aslt w s aslt    5774.439
38      (5) Rob w inj s aslt   53467.958
39      (6) Rob w inj m aslt   64188.001
40         (7) Rob wo injury   59359.504
41      (9) At rob inj m asl   10626.371
```

As you can see, there are some irregularities with the coding of crime types. Sometimes a type is coded as "(01)", but other times it is coded as "(1)". Let's standardize this coding using regular expressions.

```r
dataInc$V4529 <- gsub("\\(([1-9])\\)", "(0\\1)", dataInc$V4529)
aggregate(WGTVICDY~V4529, data=dataInc, sum)
```

```
                        V4529    WGTVICDY
1       (01) Completed rape  129132.610
2       (02) Attempted rape   61142.227
3    (03) Sex aslt w s aslt   46987.050
4    (04) Sex aslt w m aslt    6515.781
5     (05) Rob w inj s aslt  132811.230
6     (06) Rob w inj m aslt  141752.888
7        (07) Rob wo injury  235386.750
8     (08) At rob inj s asl   28969.151
9     (09) At rob inj m asl   37496.087
10       (10) At rob w aslt  148857.011
11    (11) Ag aslt w injury  385348.494
12    (12) At ag aslt w wea  271055.951
13     (13) Thr aslt w weap  421411.004
14     (14) Simp aslt w inj  954981.736
15     (15) Sex aslt wo inj   32580.327
16    (16) Unw sex wo force   15992.059
17 (17) Asl wo weap, wo inj 2005635.943
18     (18) Verbal thr rape   39745.499
19    (19) Ver thr sex aslt   15369.782
20     (20) Verbal thr aslt 2019545.074
21     (21) Purse snatching   15990.538
22     (22) At purse snatch    7272.660
23      (23) Pocket picking  126418.096
24     (31) Burg, force ent 1215286.994
25    (32) Burg, ent wo for 1758044.551
```

```
26     (33) Att force entry  711352.327
27     (40) Motor veh theft  480278.161
28    (41) At mtr veh theft  165996.837
29         (54) Theft < $10 1115139.162
30       (55) Theft $10-$49 2899929.059
31      (56) Theft $50-$249 4918627.396
32         (57) Theft $250+ 3790419.581
33      (58) Theft value NA 1369499.977
34     (59) Attempted theft  686151.735
```

Now, we can use the NCVS incident data to find out how many car thefts occurred in 2012.

```
with(subset(dataInc, V4529=="(40) Motor veh theft"),
     sum(WGTVICDY))
```

```
[1] 480278.2
```

Also, note that the definition of rape changed in 2013.

```
with(subset(dataInc,V4529=="(01) Completed rape"),
     sum(WGTVICDY))
```

```
[1] 129132.6
```

Merging in data from the household and person data

So far, we've created a dataframe and worked with weights for the Incident data. However, the Household and Person Data have data that we might need. Let's first create a 2012 data year household data frame, much like we did with the incident data. Note that `YEARQ` refers to the year and quarter of the interview. The variable `V2130` is the month allocated from panel/rotation number. The panel/rotation number refer to the process through which interviews are conducted.

```
dataHH <- rbind(dataHH12,dataHH13)
dataHH <- subset(dataHH, YEARQ>=2012.1 & YEARQ<=2013.2)
```

Let's make the "month allocated" uniform, and using regular expressions, delete "0s" following parentheses.

```
table(dataHH$V2130)
dataHH$V2130 <- gsub("\\(0", "\\(", dataHH$V2130)
table(dataHH$V2130)
```

```
  (01) January  (02) February     (03) March     (04) April        (05) May
         10602          10567          10695          10614           10511
     (06) June      (07) July    (08) August (09) September    (10) October
         10659          10572          10624          10678           10692
 (11) November  (12) December    (1) January   (2) February       (3) March
         10597          10630          10612          10573           10702
     (4) April        (5) May       (6) June       (7) July      (8) August
```

```
        10720          10661          10603              0              0
 (9) September 
            0 

  (1) January  (10) October (11) November (12) December  (2) February 
        21214         10692         10597         10630         21140 
    (3) March     (4) April       (5) May      (6) June      (7) July 
        21397         21334         21172         21262         10572 
   (8) August (9) September 
        10624         10678 
```

When you view the table again, you can see that the original 21 months listed were condensed into 12.

Next, create a 2012 data year person data frame. We need to first fix incompatible factor/numeric in 2012/2013. The factor levels in 2012 look like "(1) Yes", but in 2013 are just "1."

```
i <- sapply(dataPers12,levels)                           #gives factor levels for each variable
i <- i[!sapply(i,is.null)]                                #gives factor levels for each factor vari
                                                          #for non-factor variables, i returns a nu
i <- sapply(i, function(x) all(substring(x,1,1)=="(")) #store in i those variables where the fir
var.fix <- names(i)[i]                                    #this gives us the name of variables
                                                          #where factor levels begin with "(".

for(xj in var.fix) #create a for-loop to fix these variable names. for each value "xj" in var.fi
{
   dataPers12[,xj] <- gsub("\\(([0-9]+)\\).*","\\1",dataPers12[,xj]) #remove the words that foll
   dataPers12[,xj] <- as.numeric(dataPers12[,xj]) #convert the numbers in parentheses to just nu
}
```

Then, stack the 2012 and 2013 data frames using `rbind()`.

```
dataPers <- rbind(dataPers12, dataPers13)
dataPers <- subset(dataPers, YEARQ>=2012.1 & YEARQ<=2013.2)
```

Now that we've created a person dataframe and an incident dataframe, we can merge them together. We will use `merge()` to pull age, marital status, and sex into the incident data. The `merge()` function has several parameters that communicate to R which features should be used to match and which ones should be merged. Here we tell `merge()` to use use a pair of features from the incident data (`IDPER` and `YEARQ`) and look up a row in `dataPers` with the same values of `IDPER` and `YEARQ`. We've selected only the five columns `IDPER`, `YEARQ`, `V3014`, `V3015`, and `V3018` from `dataPers`. The first two `merge()` uses to identify matching rows and the last three will be attached as new columns to `dataInc`.

```
a <- merge(dataInc,                          # incident data
           dataPers[,c("IDPER","YEARQ",      # IDPER & YEARQ unique IDs of person
                       "V3014",              # age
                       "V3015",              # marital status
                       "V3018")],            # sex
           by=c("IDPER","YEARQ"),            # variables used to merge
           all.x=TRUE)                       # keep all incidents, even if not matched
```

```
# a should have the same number of rows as dataInc, but 3 additional new columns
dim(dataInc)
```

```
[1] 8852  951
```

```
dim(a)
```

```
[1] 8852  954
```

```
# replace dataInc with a, now containing age, marital, and sex
dataInc <- a

# check merge for first incident
dataInc[1,c("IDPER","YEARQ","V3014","V3015","V3018")]
```

```
                        IDPER  YEARQ V3014 V3015 V3018
1 250105121075958229372843501 2012.3    28     3     1
```

```
# check dataPers for this person's age, marital, and sex
subset(dataPers, IDPER=="250105121075958229372843501" & YEARQ==2012.3,
       select = c("IDPER","YEARQ","V3014","V3015","V3018"))
```

```
                            IDPER  YEARQ V3014 V3015 V3018
95199 250105121075958229372843501 2012.3    28     3     1
```

We can see that the first row of `dataInc` now has three additional columns, and that they have the correct values merged from the `dataPers` data.

Let's give these new columns better names.

```
names(dataInc)[names(dataInc)=="V3014"] <- "age"
names(dataInc)[names(dataInc)=="V3015"] <- "marital"
names(dataInc)[names(dataInc)=="V3018"] <- "sex"
```

Let's also create a new variable that breaks age into age categories.

```
dataInc$ageGroup <- cut(dataInc$age, breaks=c(0,16,21,35,45,60,110))
```

Note that "8" is a missing value indicator for marital status. Always refer to the Codebook if you are not sure what a variable or a categorical variable value means.

```
dataInc$marital[dataInc$marital==8] <- NA
```

Factor variables in R put meaningful labels on categorical variables. Instead of working with the numbers 1-5 for marital status, let's assign the number values their actual corresponding names.

```
dataInc$marital <- factor(dataInc$marital, levels=1:5,
                          labels=c("married","widowed","divorced",
                                   "separated","never married"))
dataInc$sex <- factor(dataInc$sex, levels=1:2,
                      labels=c("male","female"))
```

Let's get estimated counts by age group and sex.

```r
aggregate(WGTVICDY~ageGroup+sex, data=dataInc, FUN=sum)
```

```
   ageGroup    sex  WGTVICDY
1    (0,16]   male 1198909.6
2   (16,21]   male 1274033.7
3   (21,35]   male 3539889.7
4   (35,45]   male 2095416.6
5   (45,60]   male 3024668.5
6  (60,110]   male 1337477.9
7    (0,16] female  887078.5
8   (16,21] female 1243057.6
9   (21,35] female 4320788.8
10  (35,45] female 2307591.3
11  (45,60] female 3240564.4
12 (60,110] female 1921647.3
```

We can also find out common crime type by sex. As before, `aggregate()` will total up the weights, but as you see in the ageGroup/sex example above, `aggregate()` produces the results in a long form. Sometimes this is useful, but sometimes we want to have our results side-by-side. We will use `reshape()` to convert the "long format" results from `aggregate()` to a "wide format".

```r
a <- aggregate(WGTVICDY~V4529+sex, data=dataInc, FUN=sum)
a <- reshape(a, timevar="sex", idvar="V4529", direction="wide")
a[is.na(a)] <- 0
names(a) <- c("crimeType","male","female")
a
```

```
                  crimeType        male     female
1        (01) Completed rape    6318.130 122814.480
2        (02) Attempted rape   42077.861  19064.366
3     (03) Sex aslt w s aslt   38218.021   8769.029
4      (05) Rob w inj s aslt   80534.437  52276.793
5      (06) Rob w inj m aslt   35610.607 106142.282
6          (07) Rob wo injury 150662.017  84724.733
7      (08) At rob inj s asl   22330.349   6638.802
8      (09) At rob inj m asl   12200.917  25295.171
9          (10) At rob w aslt 104657.340  44199.671
10     (11) Ag aslt w injury 188925.090 196423.404
11     (12) At ag aslt w wea 185157.394  85898.556
12      (13) Thr aslt w weap 237527.692 183883.312
13      (14) Simp aslt w inj 448773.257 506208.479
14      (15) Sex aslt wo inj    3119.587  29460.740
15     (16) Unw sex wo force    2957.926  13034.133
16 (17) Asl wo weap, wo inj 1042741.375 962894.567
17      (18) Verbal thr rape   26408.008  13337.490
18     (19) Ver thr sex aslt    9298.262   6071.520
19      (20) Verbal thr aslt 1099721.249 919823.826
20       (23) Pocket picking   81230.111  45187.984
21      (31) Burg, force ent  609106.185 606180.810
```

```
22    (32) Burg, ent wo for  741492.194 1016552.357
23     (33) Att force entry  269383.309  441969.018
24     (40) Motor veh theft  256959.885  223318.276
25    (41) At mtr veh theft   87364.540   78632.297
26         (54) Theft < $10  444360.185  670778.978
27       (55) Theft $10-$49 1217450.179 1682478.881
28      (56) Theft $50-$249 2261589.762 2657037.634
29         (57) Theft $250+ 1825854.971 1964564.610
30      (58) Theft value NA  588405.556  781094.421
31     (59) Attempted theft  349959.481  336192.254
35   (04) Sex aslt w m aslt       0.000    6515.781
52     (21) Purse snatching       0.000   15990.538
53     (22) At purse snatch       0.000    7272.660
```

We can then convert this result to column percentages. To obtain a column percentage, we divide counts for an individual cell by the total number of counts for the column. So, the sum of all the values in the male column should equal 100:

```
temp <- a
temp$male   <- with(temp, 100*male/  sum(male))
temp$female <- with(temp, 100*female/sum(female))
colSums(temp[,-1]) # check that the columns sum to 100
```

```
  male female 
   100    100 
```

```
temp$ratio <- temp$female/temp$male
temp[order(-temp$ratio),]
```

```
                  crimeType        male      female      ratio
35   (04) Sex aslt w m aslt  0.00000000  0.04680632        Inf
52     (21) Purse snatching  0.00000000  0.11486855        Inf
53     (22) At purse snatch  0.00000000  0.05224339        Inf
1       (01) Completed rape  0.05066503  0.88224180 17.4132299
14     (15) Sex aslt wo inj  0.02501594  0.21163218  8.4598928
15    (16) Unw sex wo force  0.02371958  0.09363112  3.9474183
5     (06) Rob w inj m aslt  0.28556116  0.76247652  2.6700989
8     (09) At rob inj m asl  0.09783905  0.18170868  1.8572204
23     (33) Att force entry  2.16018250  3.17489877  1.4697364
26         (54) Theft < $10  3.56332060  4.81856254  1.3522675
27       (55) Theft $10-$49  9.76272278 12.08614160  1.2379888
22    (32) Burg, ent wo for  5.94601969  7.30243682  1.2281219
30      (58) Theft value NA  4.71841922  5.61101711  1.1891731
28      (56) Theft $50-$249 18.13566934 19.08691602  1.0524517
13     (14) Simp aslt w inj  3.59870899  3.63636503  1.0104638
29         (57) Theft $250+ 14.64151570 14.11251359  0.9638697
10    (11) Ag aslt w injury  1.51498871  1.41101390  0.9313692
21     (31) Burg, force ent  4.88441739  4.35451951  0.8915126
31     (59) Attempted theft  2.80632214  2.41504796  0.8605740
16 (17) Asl wo weap, wo inj  8.36173435  6.91698435  0.8272189
```

```
25    (41) At mtr veh theft  0.70057552  0.56485766  0.8062766
24     (40) Motor veh theft  2.06055917  1.60421408  0.7785334
19     (20) Verbal thr aslt  8.81865547  6.60758428  0.7492734
12     (13) Thr aslt w weap  1.90473257  1.32093174  0.6934998
18    (19) Ver thr sex aslt  0.07456269  0.04361496  0.5849435
4     (05) Rob w inj s aslt  0.64580498  0.37553204  0.5814945
6        (07) Rob wo injury  1.20815745  0.60862286  0.5037612
20      (23) Pocket picking  0.65138358  0.32460935  0.4983382
17     (18) Verbal thr rape  0.21176560  0.09581030  0.4524356
11    (12) At ag aslt w wea  1.48477559  0.61705507  0.4155881
2       (02) Attempted rape  0.33742202  0.13694949  0.4058700
9        (10) At rob w aslt  0.83924634  0.31750977  0.3783273
7     (08) At rob inj s asl  0.17906688  0.04769005  0.2663253
3    (03) Sex aslt w s aslt  0.30646999  0.06299260  0.2055425
```

Or we can compute row percentages to determine what percentage of each crime is male and female.

```r
temp <- a
row.total <- with(temp, male+female)
temp$male   <- with(temp, 100*male/  row.total)
temp$female <- with(temp, 100*female/row.total)
rowSums(temp[,-1]) # check that the rows sum to 100
temp$ratio <- temp$female/temp$male
temp[order(-temp$ratio),]
```

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
 19  20  21  22  23  24  25  26  27  28  29  30  31  35  52  53
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
                  crimeType      male    female      ratio
35   (04) Sex aslt w m aslt  0.000000 100.00000        Inf
52      (21) Purse snatching  0.000000 100.00000        Inf
53      (22) At purse snatch  0.000000 100.00000        Inf
1        (01) Completed rape  4.892745  95.10725 19.4384234
14      (15) Sex aslt wo inj  9.575063  90.42494  9.4437952
15    (16) Unw sex wo force 18.496217  81.50378  4.4065110
5     (06) Rob w inj m aslt 25.121609  74.87839  2.9806367
8     (09) At rob inj m asl 32.539173  67.46083  2.0732188
23      (33) Att force entry 37.869182  62.13082  1.6406696
26          (54) Theft < $10 39.847958  60.15204  1.5095389
27        (55) Theft $10-$49 41.982068  58.01793  1.3819694
22    (32) Burg, ent wo for 42.177099  57.82290  1.3709549
30      (58) Theft value NA 42.964992  57.03501  1.3274763
28      (56) Theft $50-$249 45.980099  54.01990  1.1748539
13      (14) Simp aslt w inj 46.992863  53.00714  1.1279827
29         (57) Theft $250+ 48.170260  51.82974  1.0759697
10    (11) Ag aslt w injury 49.027074  50.97293  1.0396894
21      (31) Burg, force ent 50.120357  49.87964  0.9951973
```

31	(59) Attempted theft	51.003220	48.99678	0.9606605
16	(17) Asl wo weap, wo inj	51.990561	48.00944	0.9234261
25	(41) At mtr veh theft	52.630244	47.36976	0.9000482
24	(40) Motor veh theft	53.502305	46.49770	0.8690784
19	(20) Verbal thr aslt	54.453910	45.54609	0.8364154
12	(13) Thr aslt w weap	56.364853	43.63515	0.7741553
18	(19) Ver thr sex aslt	60.497034	39.50297	0.6529736
4	(05) Rob w inj s aslt	60.638274	39.36173	0.6491235
6	(07) Rob wo injury	64.006159	35.99384	0.5623496
20	(23) Pocket picking	64.255130	35.74487	0.5562960
17	(18) Verbal thr rape	66.442765	33.55724	0.5050548
11	(12) At ag aslt w wea	68.309658	31.69034	0.4639218
2	(02) Attempted rape	68.819641	31.18036	0.4530735
9	(10) At rob w aslt	70.307297	29.69270	0.4223275
7	(08) At rob inj s asl	77.083202	22.91680	0.2972995
3	(03) Sex aslt w s aslt	81.337349	18.66265	0.2294475