

Working with National Crime Victimization Survey Data

Greg Ridgeway

Ruth Moyer

Li Sian Goh

2026-02-10

Table of contents

1	Introduction	1
2	Acquiring the NCVS data	2
3	Combining 2022 and 2023 data	16
4	BJS modifications and survey weights	22
5	Tabulating victimizations	25
6	Calculating victimization by demographics	29

1 Introduction

Through our work with NIBRS, we have already discussed reported crime. Nonetheless, not all crimes are reported to the police. Each year, under the guidance of the Bureau of Justice Statistics, the U.S. Census Bureau conducts the National Crime Victimization Survey (NCVS), a source of self-reported victimization data. The Census Bureau interviews a sample of people 12 years old or older about the number and characteristics of crime victimizations they experienced during the prior 6 months.

In 2023 226,480 people in 142,028 households participated. The survey had a 63% response rate for households and 82% response rate for individuals. Households remain in the sample for $3\frac{1}{2}$ years completing interviews every 6 months, in person or by phone, for a total of seven interviews. The survey cost \$62M annually and required roughly 125,000 hours of uncompensated respondent time.

The NCVS contains information about nonfatal personal crimes, such as rape and 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 2023 BJS Crime Victimization report](#).

2 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 data collected in 2022 and 2023 to assemble a dataset that covers victimizations occurring in 2022. Since respondents are asked about crime in the previous six months, respondents completing surveys in May 2023 will still be reporting about crimes in December 2022.

First, we will download the NCVS 2022 data, [ICPSR 38603](#). Click on Download, select R, save the resulting file (called something like ICPSR_38603-V1.zip), extract the contents of the zipped file to a convenient folder, and give it a more understandable folder name, like NCVS2022. Repeat the process for downloading the NCVS 2023 data, [ICPSR 38962](#). New NCVS data tends to appear in mid-September. Typically we need to wait about nine months to get the results from the previous year.

After unzipping the NCVS files, you will find subfolders called DS0001, DS0002, DS0003, DS0004, and DS0005.

```
list.files("NCVS2022/",recursive = TRUE)
```

```
[1] "38603-descriptioncitation.html"
[2] "38603-manifest.txt"
[3] "38603-related_literature.txt"
[4] "38603-User_guide.pdf"
[5] "DS0001/38603-0001-Codebook-ICPSR.pdf"
[6] "DS0001/38603-0001-Data.rda"
[7] "DS0002/38603-0002-Codebook-ICPSR.pdf"
[8] "DS0002/38603-0002-Data.rda"
[9] "DS0003/38603-0003-Codebook-ICPSR.pdf"
[10] "DS0003/38603-0003-Data.rda"
[11] "DS0004/38603-0004-Codebook-ICPSR.pdf"
[12] "DS0004/38603-0004-Data.rda"
[13] "DS0005/38603-0005-Codebook-ICPSR.pdf"
[14] "DS0005/38603-0005-Data.rda"
[15] "factor_to_numeric_icpsr.R"
[16] "series-95-related_literature.txt"
[17] "TermsOfUse.html"
```

```
list.files("NCVS2023/",recursive = TRUE)
```

```
[1] "38962-descriptioncitation.html"
[2] "38962-manifest.txt"
[3] "38962-related_literature.txt"
[4] "38962-User_guide.pdf"
[5] "DS0001/38962-0001-Codebook-ICPSR.epub"
[6] "DS0001/38962-0001-Codebook-ICPSR.pdf"
[7] "DS0001/38962-0001-Data.rda"
[8] "DS0002/38962-0002-Codebook-ICPSR.epub"
[9] "DS0002/38962-0002-Codebook-ICPSR.pdf"
[10] "DS0002/38962-0002-Data.rda"
[11] "DS0003/38962-0003-Codebook-ICPSR.epub"
[12] "DS0003/38962-0003-Codebook-ICPSR.pdf"
[13] "DS0003/38962-0003-Data.rda"
[14] "DS0004/38962-0004-Codebook-ICPSR.epub"
[15] "DS0004/38962-0004-Codebook-ICPSR.pdf"
[16] "DS0004/38962-0004-Data.rda"
[17] "DS0005/38962-0005-Codebook-ICPSR.epub"
[18] "DS0005/38962-0005-Codebook-ICPSR.pdf"
[19] "DS0005/38962-0005-Data.rda"
[20] "factor_to_numeric_icpsr.R"
[21] "series-95-related_literature.txt"
[22] "TermsOfUse.html"
```

Inside each of these subfolders you see an R data file with the extension `.rda`. We will spend most of our attention on the contents of the DS0005 folder, which contains the “incident-level extract file.” In each folder you will also find codebooks in pdf (and epub) format. The codebook is as important as it is tedious for understanding what is stored in the NCVS data. You should become familiar with the codebooks as soon as you can.

Let’s start loading these datasets. We will skip the DS0001 subfolder, which contains basic survey information on the targeted addresses. The DS0002 folder contains data on the households included in the survey.

```
load("NCVS2022/DS0002/38603-0002-Data.rda")
load("NCVS2023/DS0002/38962-0002-Data.rda")

# and let's give them nicer names
dataHH22 <- da38603.0002
dataHH23 <- da38962.0002
```

Take a peek at the first couple of rows

```
library(dplyr)
library(tidyr)

dataHH22 |>
  head() |>
  select(V2001, YEARQ, IDHH, V2003, V2014, V2016, V2018, V2020,
         V2030, V2031, V2032, V2034, V2036, V2038, V2040A, V2127B, V2129)
```

		V2001	YEARQ		IDHH	V2003
1	(2) Household record	2022.1	1809000258358302568236125	(221)	2022, 1st quarter	
2	(2) Household record	2022.1	1809000258380931568236135	(221)	2022, 1st quarter	
3	(2) Household record	2022.1	1809000258543680568236125	(221)	2022, 1st quarter	
4	(2) Household record	2022.1	1809000284326166568236135	(221)	2022, 1st quarter	
5	(2) Household record	2022.1	1809000284384631568236124	(221)	2022, 1st quarter	
6	(2) Household record	2022.1	1809000284459399568236114	(221)	2022, 1st quarter	
		V2014	V2016		V2018	V2020
1		<NA>	(1) Urban		<NA>	(01) House/apt/flat
2	(1) Owned/being bght	(1) Urban			<NA>	(01) House/apt/flat
3	(1) Owned/being bght	(1) Urban			<NA>	(01) House/apt/flat
4	(1) Owned/being bght	(1) Urban			<NA>	(01) House/apt/flat
5	(1) Owned/being bght	(2) Rural	(2) Less than \$1,000	(01)	House/apt/flat	
6	(1) Owned/being bght	(2) Rural	(2) Less than \$1,000	(01)	House/apt/flat	
		V2030	V2031		V2032	V2034
1	(218) Type A - Refused	(01) White only			<NA>	<NA>
2	(300) Interviewed hhld		<NA>	(11) Reference person	(2) Widowed	
3	(300) Interviewed hhld		<NA>	(11) Reference person	(3) Divorced	
4	(300) Interviewed hhld		<NA>	(11) Reference person	(3) Divorced	
5	(300) Interviewed hhld		<NA>	(11) Reference person	(1) Married	
6	(300) Interviewed hhld		<NA>	(11) Reference person	(2) Widowed	
		V2036	V2038		V2040A	V2127B
1		<NA>	<NA>		<NA>	(3) South
2	(1) Male	(42) Bachelor degree	(01) White only	(3)	South	
3	(1) Male	(42) Bachelor degree	(01) White only	(3)	South	
4	(2) Female	(43) Master degree	(01) White only	(3)	South	
5	(2) Female	(28) High school grad	(01) White only	(3)	South	
6	(2) Female	(28) High school grad	(01) White only	(3)	South	
		V2129				
1	(2) (S)MSA not city					
2	(3) Not (S)MSA					
3	(1) City of (S)MSA					

```

4 (2) (S)MSA not city
5      (3) Not (S)MSA
6 (2) (S)MSA not city

```

The 2022 household dataset has 448 columns. Instead of printing out all of them, here I just picked out 17 columns here. First off you can see that the column names are generally not helpful. That is where the codebook comes in handy. The codebook tells you what each variable means.

Somewhat hidden is a table linking column names to English explanations of what is in those columns. You can get to it by extracting the data frame's "attributes" with `attr()`.

```

varSHH <- dataHH22 |>
  attr("variable.labels") |>
  data.frame() |>
  tibble::rownames_to_column() |>
  setNames(c("varname", "details")) |>
  filter(!grepl("^HHREP", varname)) # exclude rows that start with HHREP

```

Now `varSHH` has two columns, the first with the column names and the second with the details. Let's pull up the 17 columns listed before.

```

varSHH |>
  filter(varname %in% c("V2001", "YEARQ", "IDHH", "V2003", "V2014", "V2016",
    "V2018", "V2020", "V2030", "V2031", "V2032", "V2034", "V2036",
    "V2038", "V2040A", "V2127B", "V2129"))

```

	varname	details
1	V2001	HOUSEHOLD RECORD TYPE
2	YEARQ	YEAR AND QUARTER OF INTERVIEW (YYYY.Q)
3	IDHH	NCVS ID FOR HOUSEHOLDS
4	V2003	YEAR AND QUARTER IDENTIFICATION NUMBER
5	V2014	TENURE (ORIGINAL)
6	V2016	LAND USE (ORIGINAL)
7	V2018	FARM SALES (ORIGINAL)
8	V2020	TYPE OF LIVING QUARTERS (ORIGINAL)
9	V2030	REASON FOR NONINTERVIEW
10	V2031	RACE OF HH HEAD (TYPE A NONINTERVIEW)
11	V2032	PRINCIPAL PERSON RELATION TO REF PERSON
12	V2034	PRINCIPAL PERSON MARITAL STATUS (CURR)
13	V2036	PRINCIPAL PERSON SEX (ALLOCATED)
14	V2038	PRINCIPAL PERSON EDUCATIONAL ATTAINMENT

```

15 V2040A PRINCIPAL PERSON RACE RECODE (START 2003 Q1)
16 V2127B REGION - 1990, 2000, 2010 SAMPLE DESIGN (START 1995 Q3)
17 V2129 MSA STATUS

```

These are much more intelligible descriptions. “(S)MSA” stands for the Standard Metropolitan Statistical Areas, an outdated term. Today we call them simply MSAs. Minimum population has to be 50,000, but there is movement toward redefining as 100,000.

Note the first household record has IDHH equal to 1809000258543680568236125. We can load the respondent “person file” to see who in this household responded.

```

# loading person-level data
load("NCVS2022/DS0003/38603-0003-Data.rda")
load("NCVS2023/DS0003/38962-0003-Data.rda")
dataPers22 <- da38603.0003
dataPers23 <- da38962.0003

# lookup respondents from this household
dataPers22 |>
  filter(IDHH=="1809000258543680568236125") |>
  select(!starts_with("PERREP")) # drop all the PERREP weight columns

```

	V3001	YEARQ	IDHH
1	(3) Person record	2022.1	1809000258543680568236125
2	(3) Person record	2022.3	1809000258543680568236125

	IDPER	V3002	V3003	V3004
1	180900025854368056823612501	3 (221) 2022, 1st quarter	18	
2	180900025854368056823612501	3 (223) 2022, 3rd quarter	18	

	V3005	V3006	V3008	V3009	V3010	V3011
1	09000258543680568236	1	25	1	1	(1) Personal/self
2	09000258543680568236	1	25	1	1	(2) Telephone/self

	V3012	V3013	V3014	V3015	V3016	V3017	V3018
1	(11) Reference person	62	62	(3) Divorced	(3) Divorced	(1) Male	(1) Male
2	(11) Reference person	63	63	(3) Divorced	(3) Divorced	(1) Male	(1) Male

	V3019	V3020	V3023A	V3024	V3024A	V3025	V3026
1	(2) No	(42) Bachelor degree	(01) White only	(2) No	(2) No	(02) February	12
2	(2) No	(42) Bachelor degree	(01) White only	(2) No	(2) No	(08) August	31

	V3027	V3031	V3032	V3033	V3034	V3035	V3040	V3041	V3042	V3043	V3044	V3045
1	2022	NA	23	NA	(2) No	NA	(2) No	NA	(2) No	NA	(2) No	NA
2	2022	NA	24	NA	(2) No	NA	(2) No	NA	(2) No	NA	(2) No	NA

	V3046	V3047	V3048	V3049	V3050	V3051	V3052	V3053	V3054	V3055	V3056	V3057
1	(2) No	NA	(2) No	<NA>	<NA>	<NA>	<NA>	NA	(2) No	<NA>	<NA>	<NA>

2	(2)	No	NA	(2)	No	<NA>	<NA>	<NA>	<NA>	NA	(2)	No	<NA>	<NA>	<NA>
		V3058	V3059				V3060	V3061	V3062	V3063	V3064	V3065	V3066		
1		<NA>	NA	(1)	At least 1 entry	(0)	No	(1)	Yes	(0)	No	(0)	No	(0)	No
2		<NA>	NA	(1)	At least 1 entry	(1)	Yes	(0)	No	(0)	No	(0)	No	(0)	No
		V3067	V3068				V3069	V3070	V3_V4526H3A	V3_V4526H3B	V3_V4526H5				
1	(0)	No	(0)	No	(0)	No out of range	<NA>		(2)	No	(2)	No	(2)	No	
2	(0)	No	(0)	No	(0)	No out of range	<NA>		(2)	No	(2)	No	(2)	No	
		V3_V4526H4	V3_V4526H6	V3_V4526H7								V3083			
1		(2)	No	(2)	No	(2)	No	(1)	Yes, born in the United States						
2		(2)	No	(2)	No	(2)	No	(1)	Yes, born in the United States						
							V3084	V3085	V3086						
1	(2)	Straight, that is, not lesbian or gay	(1)	Male	(1)	Male									
2	(2)	Straight, that is, not lesbian or gay	(1)	Male	(1)	Male									
							V3087	V3088	V3089	V3090	V3091	V3092	V3093	V3094	
1	(1)	Never served in the military	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	
2	(1)	Never served in the military	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	
		V3097A	V3098	V3071	V3072	V3073			V3074			V3075			
1	<NA>	<NA>	(1)	Yes	<NA>	<NA>	(27)	Something else	(3)	St/cnty/loc govt					
2	<NA>	<NA>	(1)	Yes	<NA>	<NA>	(27)	Something else	(3)	St/cnty/loc govt					
		V3076		V3077	V3078				V3079	V3080	WGTPERCY				
1	(1)	A city	(1)	Hhld resp	(2)	No	(4)	None above schools	1207.963	603.9816					
2	(1)	A city	(1)	Hhld resp	(2)	No	(4)	None above schools	1470.257	735.1283					
		V3081	V3082	PER_TIS	PERINTVNUM	PINTTYPE_TIS1				PINTTYPE_TIS2					
1	0	2022		5		5		<NA>	(1)	Personal, Self-respondent					
2	0	2022		6		6		<NA>	(1)	Personal, Self-respondent					
				PINTTYPE_TIS3						PINTTYPE_TIS4					
1	(2)	Telephone, Self-respondent	(2)	Telephone, Self-respondent											
2	(2)	Telephone, Self-respondent	(2)	Telephone, Self-respondent											
				PINTTYPE_TIS5						PINTTYPE_TIS6					
1	(2)	Telephone, Self-respondent	(1)	Personal, Self-respondent											
2	(2)	Telephone, Self-respondent	(1)	Personal, Self-respondent											
				PINTTYPE_TIS7						PERBOUNDED					
1				<NA>	(1)	Bounded by previous time in sample									
2	(2)	Telephone, Self-respondent	(1)	Bounded by previous time in sample											

These two rows represent two surveys, six months apart of the same divorced, 62-63 year-old, white male. Let's look up another household.

```
dataPers22 |>
  filter(IDHH=="1809000284384631568236124") |>
  select(!starts_with("PERREP"))
```

V3001 YEARQ

IDHH

1 (3) Person record 2022.1 1809000284384631568236124
2 (3) Person record 2022.1 1809000284384631568236124

	IDPER	V3002					V3003	V3004
1	180900028438463156823612401	5	(221)	2022,	1st	quarter	18	
2	180900028438463156823612402	5	(221)	2022,	1st	quarter	18	
		V3005	V3006	V3008	V3009	V3010	V3011	
1	09000284384631568236	1	24	1	1	(2) Telephone/self		
2	09000284384631568236	1	24	2	2	(5) Noninterview		
		V3012	V3013	V3014	V3015	V3016	V3017	
1	(11) Reference person	69	69	(1) Married	(1) Married	(2) Female		
2	(01) Husband	60	60	(1) Married	(1) Married	(1) Male		
		V3018	V3019	V3020	V3023A	V3024	V3024A	
1	(2) Female	<NA>	(28) High school grad	(01) White only	(2) No	(2) No		
2	(1) Male	(2) No	(28) High school grad	(01) White only	(2) No	(2) No		
		V3025	V3026	V3027	V3031	V3032	V3033	V3034
1	(02) February	18	2022	NA	9	NA	(2) No	NA
2	<NA>	NA	NA	NA	NA	NA	<NA>	NA
		V3043	V3044	V3045	V3046	V3047	V3048	V3049
1	NA	(2) No	NA	(2) No	NA	(2) No	<NA>	<NA>
2	NA	<NA>	NA	<NA>	NA	<NA>	<NA>	<NA>
		V3055	V3056	V3057	V3058	V3059	V3060	V3061
1	<NA>	<NA>	<NA>	<NA>	NA	(1) At least 1 entry	(1) Yes	(0) No
2	<NA>	<NA>	<NA>	<NA>	NA	<NA>	<NA>	<NA>
		V3064	V3065	V3066	V3067	V3068	V3069	V3070
1	(0) No	(0) No	(0) No	(0) No	(0) No	(0) No	out of range	<NA>
2	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
		V3_V4526H3B	V3_V4526H5	V3_V4526H4	V3_V4526H6	V3_V4526H7		
1	(2) No	(2) No	(2) No	(2) No	(2) No	(2) No		
2	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>		
		V3083					V3084	
1	(1) Yes, born in the United States	(2) Straight, that is, not lesbian or gay						
2	<NA>	<NA>						
		V3085	V3086	V3087	V3088	V3089	V3090	
1	(2) Female	(2) Female	(1) Never served in the military	<NA>	<NA>	<NA>		
2	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>		
		V3091	V3092	V3093	V3094	V3097A	V3098	V3071
1	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	(1) Yes	<NA>
2	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
		V3075	V3076	V3077	V3078			
1	(1) Priv company	(3) Rural area	(1) Hhld resp	(2) No				
2	<NA>	<NA>	(0) Not hhld resp	<NA>				
		V3079	V3080	WGTPERCY	V3081	V3082	PER_TIS	PERINTVNUM
1	(4) None above schools	1070.377	535.1885	0	2022	7	7	

2	(4) None above schools	0.000	0.0000	0	2022	7	6
	PINTTYPE_TIS1					PINTTYPE_TIS2	
1	(2) Telephone, Self-respondent	(2)	Telephone, Self-respondent				
2	(4) Telephone, Proxy	(4)	Telephone, Proxy				
	PINTTYPE_TIS3					PINTTYPE_TIS4	
1	(2) Telephone, Self-respondent	(2)	Telephone, Self-respondent				
2	(4) Telephone, Proxy	(4)	Telephone, Proxy				
	PINTTYPE_TIS5					PINTTYPE_TIS6	
1	(2) Telephone, Self-respondent	(2)	Telephone, Self-respondent				
2	(2) Telephone, Self-respondent	(2)	Telephone, Self-respondent				
	PINTTYPE_TIS7					PERBOUNDED	
1	(2) Telephone, Self-respondent	(1)	Bounded by previous time in sample				
2	(5) Noninterview	(1)	Bounded by previous time in sample				

These rows represent two surveys occurring at the same time, one of the reference person, a married white female, and a second survey of her husband.

Let's grab the variable details as we did with the household data.

```
# Person file also has list of variable details
varsPers <- dataPers22 |>
  attr("variable.labels") |>
  data.frame() |>
  tibble::rownames_to_column() |>
  setNames(c("varname", "details")) |>
  filter(!grepl("^(PERREP|PINTTYPE)", varname))
varsPers
```

	varname
1	V3001
2	YEARQ
3	IDHH
4	IDPER
5	V3002
6	V3003
7	V3004
8	V3005
9	V3006
10	V3008
11	V3009
12	V3010
13	V3011

14	V3012
15	V3013
16	V3014
17	V3015
18	V3016
19	V3017
20	V3018
21	V3019
22	V3020
23	V3023A
24	V3024
25	V3024A
26	V3025
27	V3026
28	V3027
29	V3031
30	V3032
31	V3033
32	V3034
33	V3035
34	V3040
35	V3041
36	V3042
37	V3043
38	V3044
39	V3045
40	V3046
41	V3047
42	V3048
43	V3049
44	V3050
45	V3051
46	V3052
47	V3053
48	V3054
49	V3055
50	V3056
51	V3057
52	V3058
53	V3059
54	V3060
55	V3061
56	V3062

57	V3063
58	V3064
59	V3065
60	V3066
61	V3067
62	V3068
63	V3069
64	V3070
65	V3_V4526H3A
66	V3_V4526H3B
67	V3_V4526H5
68	V3_V4526H4
69	V3_V4526H6
70	V3_V4526H7
71	V3083
72	V3084
73	V3085
74	V3086
75	V3087
76	V3088
77	V3089
78	V3090
79	V3091
80	V3092
81	V3093
82	V3094
83	V3097A
84	V3098
85	V3071
86	V3072
87	V3073
88	V3074
89	V3075
90	V3076
91	V3077
92	V3078
93	V3079
94	V3080
95	WGTPERCY
96	V3081
97	V3082
98	PER_TIS
99	PERINTVNUM

100 PERBOUNDED

1	PERSON RECORD
2	YEAR AND QUARTER OF INTERVIEW
3	NCVS ID FOR HOUSEHOLD
4	NCVS ID FOR PERSON
5	ICPSR HOUSEHOLD IDENTIFICATION
6	YEAR AND QUARTER IDENTIFICATION
7	SAMPLE
8	SCRAMBLED CONTROL
9	HOUSEHOLD
10	PANEL AND ROTATION
11	PERSON SEQUENCE
12	PERSON LINE
13	TYPE OF INTERVIEW
14	RELATIONSHIP TO REFERENCE
15	AGE (ORIGINAL)
16	AGE (ALL)
17	MARITAL STATUS (CURRENT)
18	MARITAL STATUS (PREVIOUS)
19	SEX (ORIGINAL)
20	SEX (ALL)
21	NOW AN ARMED FORCE
22	EDUCATIONAL ATTAINMENT
23	RACE RECODE (START 2)
24	HISPANIC ORIGIN
25	HISPANIC ORIGIN (ALLOCATED) (START 2)
26	MONTH INTERVIEW CONDUCTED
27	DAY INTERVIEW CONDUCTED
28	YEAR INTERVIEW CONDUCTED
29	HOW LONG AT THIS ADDRESS
30	HOW LONG AT THIS ADDRESS
31	HOW MANY TIMES MOVED IN LAST
32	SOMETHING STOLEN OR
33	NO. TIMES SOMETHING STOLEN OR
34	ATTACK, THREAT, THEFT: LOCATED
35	NO. TIMES ATTACK, LOCATED
36	ATTACK, THREAT: WEAPON & ATT
37	NO. TIMES ATTACK, WEAPON
38	STOLEN, ATTACK, THREAT: OFFENDER
39	NO. TIMES ATTACK, OFFENDER
40	FORCED OR COERCED UNWARRANTED
41	NO. TIMES UNWARRANTED

42 CALL POLICE TO REPORT SOMETHING
 43 FIRST
 44 SECOND
 45 THIRD
 46 CHECK B: ATTACK, THREATEN
 47 NO. TIMES ATTACK, THREATEN
 48 THOUGHT CRIME BUT DIDN'T CALL
 49 FIRST
 50 SECOND
 51 THIRD
 52 CHECK C: ATTACK, THREATEN
 53 NO. TIMES ATTACK, THREATEN
 54 LI WHO PRESENT DURING SCREEN Q
 55 C TELEPHONE ID
 56 C NO ONE BESIDES RESPONDENT
 57 C RESPONDENT'S
 58 C HH MEMBER(S) 12+, NO
 59 C HH MEMBER(S) U
 60 C NONHOUSEHOLD M
 61 C SOMEONE PRESENT, CAN'T
 62 C DON'T KNOW IF SOMEONE ELSE
 63 RESIDUE: WHO PRESENT DURING
 64 DID SELECTED RESPONDENT HE
 65 ARE YOU DEAF OR DO YOU HAVE SERIOUS DIFFICULTY HEARING? (START 2
 66 ARE YOU BLIND OR DO YOU HAVE SERIOUS DIFFICULTY SEEING EVEN WHEN WEARING GLASSES (START 2
 67 DIFFICULT: LEARN, REMEMBER, CONCENTRATE (START 2
 68 LIMITS PHYSICAL ACTIVITIES (START 2
 69 DIFFICULT: DRESSING, BATHING, GET AROUND HOME (START 2
 70 DIFFICULT: GO OUTSIDE HOME TO SHOP OR DR OFFICE (START 2
 71 CITIZENSHIP STATUS (START 2
 72 SEXUAL ORIENTATION (START 2
 73 GENDER IDENTITY AT BIRTH (START 2
 74 CURRENT GENDER IDENTITY (START 2
 75 SERVE ON ACTIVE DUTY (START 2
 76 LI: WHEN ON ACTIVE DUTY (START 2
 77 C ACTIVE DUTY: SEPTEMBER 2001 (START 2
 78 C ACTIVE DUTY: AUGUST 1990 TO AUGUST 2001 (START 2
 79 C ACTIVE DUTY: MAY 1975 TO JULY 1990 (START 2
 80 C ACTIVE DUTY: VIETNAM ERA (AUGUST 1964 TO APRIL 1975) (START 2
 81 C ACTIVE DUTY: FEBRUARY 1955 TO JULY 1964 (START 2
 82 C ACTIVE DUTY: KOREAN WAR (JULY 1950 TO JANUARY 1955) (START 2
 83 C ACTIVE DUTY: DISCLOSURE RECODE (START 2
 84 RESIDUE: ACTIVE DUTY (START 2

```

85                                     HAVE JOB OR WORK L
86                                     HAVE JOB OR WORK IN LAST 6
87                                     DID JOB/WORK LAST 2 WEEKS
88                                     WHICH BEST DESCRIBES Y
89                                     IS EMPLOYMENT PRIVATE, GOVT
90                                     IS WORK MOSTLY IN CITY, SUBURB
91                                     HOUSEHOLD RE
92                                     EMPLOYED BY A COLLEGE OR UNIV
93                                     ATTENDING
94                                     PERSON
95                                     ADJUSTED PERSON WEIGHT - COLLECTED
96                                     NUMBER OF CRIME INCIDENTS
97                                     YEAR IDENTIFICATION (START YEAR)
98                                     PERSON TIME IN SAMPLE (START YEAR)
99                                     PERSON INTERVIEW NUMBER (START YEAR)
100                                    PERSON BOUNDED BY PREVIOUS TIME IN SAMPLE (START YEAR)

```

There is an incident-level file that we will read in here. We are not going to look at it further, since much of the information in this file is also in the incident extract file.

```

load("NCVS2022/DS0004/38603-0004-Data.rda")
load("NCVS2023/DS0004/38962-0004-Data.rda")
dataInc22 <- da38603.0004
dataInc23 <- da38962.0004

dataInc22 |>
  select(IDHH, IDPER, V4014, V4529) |>
  head()

```

	IDHH	IDPER	V4014
1	1809010265731899564536114	180901026573189956453611401	(12) December
2	1809040225522254568236115	180904022552225456823611501	(11) November
3	1809213903398449563644234	180921390339844956364423401	(09) September
4	1809240299163750563236135	180924029916375056323613501	(09) September
5	1809243565469154563238115	180924356546915456323811501	(12) December
6	1809243958169129563244125	180924395816912956324412501	(08) August

	V4529
1	(31) Burg, force ent
2	(56) Theft \$50-\$249
3	(56) Theft \$50-\$249
4	(56) Theft \$50-\$249
5	(58) Theft value NA

6 (20) Verbal thr aslt

Finally, we will load in the incident extract file and its associated variable details. This extract file merges in household-level and person-level information to the incident-level file, allowing you to connect person-level features with features of the victimizations they report.

```
# incident-level extract file
load("NCVS2022/DS0005/38603-0005-Data.rda")
load("NCVS2023/DS0005/38962-0005-Data.rda")
dataExt22 <- da38603.0005
dataExt23 <- da38962.0005

varsExt <- dataExt22 |>
  attr("variable.labels") |>
  data.frame() |>
  tibble::rownames_to_column() |>
  setNames(c("varname", "details")) |>
  filter(!grepl("INCREPWGT|VICREPWGT", varname))
```

Let's take a look at a few of the reported crime victimizations. Here I will just pull the respondent's age, marital status, sex, general location, and crime type.

```
dataExt22 |>
  select(V3014, V3015, V3018, V4022, V4529) |>
  slice(1:3)
```

	V3014	V3015	V3018	V4022	V4529
1	56 (3) Divorced	(2) Female	(3) Same city etc	(31) Burg, force ent	
2	78 (2) Widowed	(2) Female	(3) Same city etc	(56) Theft \$50-\$249	
3	43 (1) Married	(1) Male	(3) Same city etc	(56) Theft \$50-\$249	

Not all information from the household and person files are in the extract file, but many of the features that are likely to be of interest are there.

Now that the datasets are loaded and renamed, we can remove objects from our working environment that we no longer need. We can use `rm()` to accomplish this.

```
rm(da38603.0002, da38603.0003, da38603.0004, da38603.0005,
   da38962.0002, da38962.0003, da38962.0004, da38962.0005)
```

3 Combining 2022 and 2023 data

Here we are going to create a data frame containing all the reported incidents that *occurred* in 2022. Take a look at the month and year of the reported crime incidents.

```
dataExt22 |> count(V4015, V4014)
```

	V4015	V4014	n
1	2021	(07) July	138
2	2021	(08) August	280
3	2021	(09) September	384
4	2021	(10) October	510
5	2021	(11) November	650
6	2021	(12) December	849
7	2022	(01) January	772
8	2022	(02) February	727
9	2022	(03) March	774
10	2022	(04) April	730
11	2022	(05) May	770
12	2022	(06) June	839
13	2022	(07) July	718
14	2022	(08) August	593
15	2022	(09) September	383
16	2022	(10) October	292
17	2022	(11) November	136

```
dataExt23 |> count(V4015, V4014)
```

	V4015	V4014	n
1	2022	(07) July	175
2	2022	(08) August	253
3	2022	(09) September	364
4	2022	(10) October	524
5	2022	(11) November	674
6	2022	(12) December	859
7	2023	(01) January	753
8	2023	(02) February	681
9	2023	(03) March	699
10	2023	(04) April	688
11	2023	(05) May	738
12	2023	(06) June	760

```

13 2023      (07) July 721
14 2023      (08) August 573
15 2023 (09) September 447
16 2023      (10) October 273
17 2023      (11) November 142

```

Note that the 2022 NCVS reports on crimes that occurred in 2022 and 2021. Similarly, the NCVS 2023 reports on crimes that occurred in 2023 and 2022. Remember that the NCVS surveys respondents about any victimizations from the prior 12 months. We are going to stack the 2022 and 2023 incident extract data frames and then filter it to exclude 2021 and 2023.

`bind_rows()` stacks data frames on top of each other, useful when combining two datasets that have the same structure. First we will check that they have the same columns in them.

```
identical(names(dataExt22), names(dataExt23))
```

```
[1] TRUE
```

Good so far! Now let's try to stack them.

```
dataExt <- dataExt22 |>
  bind_rows(dataExt23)
```

```
Error in `bind_rows()` :
! Can't combine `..1$V2061` <factor<f6015>> and `..2$V2061` <double>.
```

Hmmm... R is complaining about V2061. Note that it specifically complains that one data frame has V2061 stored as a factor (a categorical variable) and the other one has it stored as a double, a decimal number.

```
dataExt22 |> count(V2061)
```

```

  V2061      n
1 (01) 1      8
2  <NA> 9537

```

```
dataExt23 |> count(V2061)
```

	V2061	n
1	1	14
2	3	1
3	4	3
4	NA	9306

What is V2061 anyway?

```
varsExt |> filter(varname=="V2061")
```

	varname	details
1	V2061	LINE NO. OF 4TH PROXY RESPONDENT

This reports on who reported on behalf of an unavailable respondent. Not really important for us so let's drop this one by using `select(-V2061)` on both data frames.

```
dataExt <- dataExt22 |>
  select(-V2061) |>
  bind_rows(dataExt23 |>
    select(-V2061))
```

```
Error in `bind_rows()` :
! Can't combine `..1$V4126` <factor<fb04b>> and `..2$V4126` <double>.
```

Ughh. Now it is complaining about V4126.

```
varsExt |> filter(varname=="V4126")
```

	varname	details
1	V4126	WHICH INJURY FROM OTHER WEAPON (3RD)

```
dataExt22 |> count(V4126)
```

	V4126	n
1	(10) Bruises, cuts	3
2	<NA>	9542

```
dataExt23 |> count(V4126)
```

```

  V4126      n
1    NA 9324

```

In the codebook we can find the full question: “Q.33.3 Which injuries were caused by a weapon OTHER than a gun or knife?”. This seems like a potentially interesting question that I probably do not want to discard. The issue is that no 2023 respondent said there was a third weapon that injured them. In 2022 V4126 was stored as a factor, but in 2023, since they are all missing, R defaulted to numeric (double). We can fix this by just telling R to convert the 2023 data into a factor.

```

dataExt <- dataExt22 |>
  select(-V2061) |>
  bind_rows(dataExt23 |>
    select(-V2061) |>
    mutate(V4126=as.factor(V4126)))

```

```

Error in `bind_rows()` :
! Can't combine `..1$V4313` <factor<514cc>> and `..2$V4313` <double>.

```

Dammit! Now it is complaining about V4313. What is the problem with this one? Again we have a problem with 2022 storing as a factor and 2023 storing as a double.

```
varsExt |> filter(varname=="V4313")
```

```

  varname                                details
1  V4313 3RD LINE NO. OF OTHER OWNER THEFT ITEMS

```

```
dataExt22 |> count(V4313)
```

```

  V4313      n
1 (06) 6      2
2  <NA> 9543

```

```
dataExt23 |> count(V4313)
```

	V4313	n
1	4	1
2	5	1
3	NA	9322

This column answers “Besides the respondent, which household member(s) owned the (property/money) the offender tried to take?” In 2022, the only responses were missing or #6. In 2023, the responses were missing, 4, or 5. These should be numbers since they are supposed to link respondents in the same household affected by the theft. The approach I’ll take is to use `case_match()` telling R to change the 2022 “(006) 6” response to a regular 6.

```
dataExt <- dataExt22 |>
  select(-V2061) |>
  mutate(V4313 = case_match(V4313,
                            "(06) 6" ~ 6,
                            .default = NA)) |>

  bind_rows(dataExt23 |>
    select(-V2061) |>
    mutate(V4126=as.factor(V4126)))
```

Error in `bind_rows()`:
! Can't combine `..1\$V4357A` <double> and `..2\$V4357A` <factor<758b7>>.

Will this ever stop?!?!? Another double in one year and factor in another year, this time affecting V4357A asking about handguns.

```
varsExt |> filter(varname=="V4357A")
```

	varname	details
1	V4357A	HOW MANY HANDGUNS WERE TAKEN (START 2004 Q1)

```
dataExt22 |> count(V4357A)
```

	V4357A	n
1	1	41
2	2	8
3	3	3
4	997	1
5	998	1
6	NA	9491

```
dataExt23 |> count(V4357A)
```

	V4357A	n
1	(001) 1	48
2	(998) Residue	1
3	<NA>	9275

You might have seen this word “Residue” show up before. For the NCVS, BJS records “Residue” when there is a data entry error resulting in an out-of-range code, an incorrect or unusable answer by the respondent, or the absence of an entry for a question that should have been asked. Sometimes you might also see “Out of universe/blank.” This happens when a value is outside the range of questions to be answered. For example, “Received Medical Care for Injury,” only victims who report being injured are asked whether they received medical care. All other victims skip this question.

I will solve this issue by recoding the 2023 values to numeric values.

```
dataExt <- dataExt22 |>
  select(-V2061) |>
  mutate(V4313 = case_match(V4313,
                            "(06) 6" ~ 6,
                            .default = NA)) |>
  bind_rows(dataExt23 |>
    select(-V2061) |>
    mutate(V4126 = as.factor(V4126),
           V4357A = case_match(V4357A,
                                "(001) 1" ~ 1,
                                "(998) Residue" ~ 998,
                                .default=NA)))
```

Success! Now let’s check that all is okay now.

```
dataExt |> count(V4126)
```

	V4126	n
1	(10) Bruises, cuts	3
2	<NA>	18866

```
dataExt |> count(V4313)
```

	V4313	n
1	4	1
2	5	1
3	6	2
4	NA	18865

```
dataExt |> count(V4357A)
```

	V4357A	n
1	1	89
2	2	8
3	3	3
4	997	1
5	998	2
6	NA	18766

Remember that we still have data in here from 2021 and 2023.

```
table(dataExt$V4015)
```

2021	2022	2023
2811	9583	6475

We are just going to focus on 2022.

```
dataExt <- dataExt |> filter(V4015==2022)
```

Note that BJS official reports generally classify by the year of the survey and not by the year of the crime.

4 BJS modifications and survey weights

Some respondents report crime victimizations that occurred outside of the United States.

```
# V4022 - IN WHAT CITY, TOWN, VILLAGE.
dataExt |> count(V4022)
```

	V4022	n
1	(1) Outside U.S.	44
2	(2) Not in city etc	67
3	(3) Same city etc	8069
4	(4) Diff city etc	1354
5	(5) Don't know	37
6	(8) Residue	12

The BJS convention is to exclude these crimes in official reports (see 2023 User Guide, page 21).

```
dataExt <- dataExt |>
  filter(is.na(V4022) | V4022!="(1) Outside U.S.")
```

Some crimes happen in a series. For example, a respondent may report on regular domestic abuse that happened numerous times over the last six months. Each incident of domestic abuse is a victimization, but the BJS convention is to include up to 10 occurrences for crimes reported as a series (2023 User Guide, pages 18-19).

Variable V4016 records the answer to “Altogether, how many times did this type of incident happen during the last 6 months?” and variable V4019 documents “Can you (respondent) recall enough details of each incident to distinguish them from each other?”

Note that the coding of V4016 has 997 representing “Don’t know” and 998 representing “Residue”. These are not counts of victimizations. The logic in the `case_when()` statement below checks for counts between 11 and 996 and sets the value of V4016 to 10 in that case.

```
dataExt <- dataExt |>
  mutate(V4016 = case_when(
    V4019=="(2) No (is series)" & V4016>=11 & V4016<=996 ~ 10,
    V4016 >= 997 ~ NA,
    .default=V4016))
```

The NCVS sampling design oversamples respondents in places more likely to have crime victimization. This makes the sampling effort more efficient. Otherwise, a purely random sample would contact a lot of people who had no victimization to report. As a result, the raw data from the NCVS do not reflect crime victimization in the United States. We must use the NCVS sampling weights to undo the oversampling of crime victims.

Constructing the sampling weights is a complex process (see the User Guide “Weights Details” section starting on Page 22). The NCVS sampling weights adjusts for six factors. From the User Guide:

1. *Base weight*: The inverse of the national sampling rate for the stratum of that unit (person or household).
2. *Weighting control*: Adjusts for any sub-sampling due to unexpected events in the field, such as new construction, area segments larger than anticipated, and other deviations from the overall stratum sampling rate.
3. *Household non-interview adjustment*: Adjusts for nonresponse at the household-level by increasing the weights of interviewed households most similar to households not interviewed in terms of race, MSA status of residence, and urban/suburban/rural status of residence. This inflates the weight value assigned to interviewed households so that they represent themselves and non-interviewed households. The non-interviewed cases are assigned a weight of zero, thereby excluding them from population estimates.
4. *Within-household non-interview adjustment*: Adjusts for non-response at the person-level by increasing the weight of interviewed persons most similar to persons not interviewed in terms of region, age, race, sex, and household composition. The adjustment inflates the weight value assigned to completed interviews, so that they represent themselves and sampled individuals who were not interviewed. The non-interviewed cases are assigned a weight of zero.
5. *First stage ratio estimates factor*: Adjusts for differences between characteristics of the sample non-self-representing (NSR) primary sampling units (PSUs) and independent measures of the population NSR PSUs. (For self-representing PSUs this factor is set to 1). This factor adjusts for PSU differences on region, MSA status, urban/suburban/rural status, and racial composition.
6. *Second stage ratio estimate factor*: A post-stratification factor defined for each person to adjust for the difference between weighted counts of persons (using the above five weight components) and independent estimates of the number of persons, within certain age by race by sex categories. These independent estimates are based on the Census population controls adjusted for the undercount.

Fortunately for us, the variable `SERIES_WEIGHT` captures all these adjustments and contains the weight that BJS uses for its official reports. It includes the adjustment for capping series crimes at 10.

 Always use the sampling weights

Importantly, every calculation you do with the NCVS must involve the weights. This includes weighted means, weighted percentages, and weighted counts. Even plots and figures should use the weights.

Where you would normally compute a sample mean as $\frac{\sum x_i}{n}$, the weighted mean is

$$\frac{\sum w_i x_i}{\sum w_i}$$

For a weighted percentage, total all the weights for respondents with the particular feature divided by the total weight. For some plots there is not an obvious way to accommodate the sampling weights. In those cases we can sample with replacement with probabilities proportional to the sampling weights and plot the sampled points.

5 Tabulating victimizations

First, we need to be clear about what we are counting. BJS will report on *victimizations* and *incidents*. Victimizations count the number of times a US person was victimized. Incidents count the number of times a crime incident occurred and those incidents could involve multiple victims. BJS reports largely focus on criminal victimizations.

We can start by just asking the NCVS data how many criminal victimizations there were in 2022. We compute that as the sum of all the weights.

```
sum(dataExt$SERIES_WEIGHT)
```

```
[1] 20121320
```

This means that the NCVS estimates that there were 20,121,320 criminal victimizations in the United States in 2022.

Let's take a closer look at what kinds of victimization occurred. Note that this code breaks the dataset into groups based on the reported crime type, **V4529**, and computes the total weight associated with each of those crime categories.

```
dataExt |>
  group_by(V4529) |> # crime type
  summarize(total = sum(SERIES_WEIGHT)) |>
  print(n=Inf)
```

```
# A tibble: 33 x 2
  V4529                total
  <fct>              <dbl>
1 (01) Completed rape 127522.
2 (02) Attempted rape  67330.
```

3	(03)	Sex aslt w s aslt	27038.
4	(04)	Sex aslt w m aslt	7511.
5	(05)	Rob w inj s aslt	87026.
6	(06)	Rob w inj m aslt	88620.
7	(07)	Rob wo injury	216589.
8	(08)	At rob inj s asl	25197.
9	(09)	At rob inj m asl	47896.
10	(10)	At rob w aslt	119279.
11	(11)	Ag aslt w injury	449842.
12	(12)	At ag aslt w wea	397249.
13	(13)	Thr aslt w weap	637952.
14	(14)	Simp aslt w inj	578068.
15	(15)	Sex aslt wo inj	118078.
16	(16)	Unw sex wo force	52425.
17	(17)	Asl wo weap, wo inj	1295757.
18	(18)	Verbal thr rape	41862.
19	(19)	Ver thr sex aslt	15501.
20	(20)	Verbal thr aslt	1978623.
21	(21)	Purse snatching	35375.
22	(23)	Pocket picking	124503.
23	(31)	Burg, force ent	549203.
24	(32)	Burg, ent wo for	1074214.
25	(33)	Att force entry	312530.
26	(40)	Motor veh theft	518797.
27	(41)	At mtr veh theft	193058.
28	(54)	Theft < \$10	623035.
29	(55)	Theft \$10-\$49	1735619.
30	(56)	Theft \$50-\$249	2968819.
31	(57)	Theft \$250+	3057945.
32	(58)	Theft value NA	1799273.
33	(59)	Attempted theft	749585.

The first 20 crime types listed are the violent crimes and the remainder are property crimes. Let's extract the two-digit code between the parentheses so that we can classify crime types as violent or property. First, I will run a little test code to make sure my regular expression and the crime type classification works correctly.

```
dataExt |>
  mutate(crimeCode = gsub("([0-9][0-9]).*", "\\1", V4529),
         crimeType = ifelse(crimeCode <= 20, "violent", "property")) |>
  select(V4529, crimeCode, crimeType) |>
  head()
```

	V4529	crimeCode	crimeType
1	(41) At mtr veh theft	41	property
2	(59) Attempted theft	59	property
3	(58) Theft value NA	58	property
4	(57) Theft \$250+	57	property
5	(55) Theft \$10-\$49	55	property
6	(07) Rob wo injury	07	violent

That all looks correct, so now I can move on to the tabulation.

```
a <- dataExt |>
  mutate(crimeCode = gsub("([0-9][0-9]).*", "\\1", V4529),
         crimeType = ifelse(crimeCode <= 20, "violent", "property")) |>
  group_by(crimeType) |>
  summarize(estTotal = sum(SERIES_WEIGHT))
a
```

```
# A tibble: 2 x 2
  crimeType estTotal
  <chr>      <dbl>
1 property 13741956.
2 violent  6379364.
```

In 2022, there was an estimated 6,379,364 violent crimes and 13,741,956 property crimes.

We can summarize other categories, like car thefts, attempted and completed.

```
dataExt |>
  filter(V4529 %in% c("(40) Motor veh theft",
                     "(41) At mtr veh theft")) |>
  summarize(sum(SERIES_WEIGHT))
```

```
sum(SERIES_WEIGHT)
1          711855.4
```

Measuring sexual assault has been complicated by numerous, sometimes major, changes (improvements, more precisely) in the definitions and data collection methods (e.g. question wording). The Uniform Crime Report made a major change to the [definition of rape changed](#) in 2013. The NCVS's most recent change was in 2024. See Fisher and Gross (2025) for an extended discussion and timeline of the changes.

Here is the NCVS estimate of the number of sexual assaults (attempted and completed) for 2022.

```
dataExt |>
  filter(V4529 %in% c("(01) Completed rape",
                     "(02) Attempted rape")) |>
  summarize(sum(SERIES_WEIGHT))
```

```
      sum(SERIES_WEIGHT)
1      194852.7
```

NCVS official reports combine all rape and sexual assaults. There are a lot of crime categories that describe sexual assaults.

```
unique(dataExt$V4529)
```

```
[1] (41) At mtr veh theft      (59) Attempted theft      (58) Theft value NA
[4] (57) Theft $250+          (55) Theft $10-$49        (07) Rob wo injury
[7] (13) Thr aslt w weap      (17) Asl wo weap, wo inj (20) Verbal thr aslt
[10] (32) Burg, ent wo for     (56) Theft $50-$249      (14) Simp aslt w inj
[13] (54) Theft < $10         (31) Burg, force ent     (19) Ver thr sex aslt
[16] (23) Pocket picking      (10) At rob w aslt       (11) Ag aslt w injury
[19] (40) Motor veh theft     (33) Att force entry     (12) At ag aslt w wea
[22] (04) Sex aslt w m aslt   (21) Purse snatching     (03) Sex aslt w s aslt
[25] (16) Unw sex wo force    (09) At rob inj m asl    (02) Attempted rape
[28] (06) Rob w inj m aslt    (15) Sex aslt wo inj     (18) Verbal thr rape
[31] (01) Completed rape      (08) At rob inj s asl    (05) Rob w inj s aslt
34 Levels: (01) Completed rape (02) Attempted rape ... (59) Attempted theft
```

Let's count any that have the word "rape" and any that have the word "sex" (sometimes capitalized). Here are the ones that BJS counts in this category.

```
dataExt |>
  filter(grepl("rape|[Ss]ex", V4529)) |>
  distinct(V4529)
```

```
      V4529
1 (19) Ver thr sex aslt
2 (04) Sex aslt w m aslt
3 (03) Sex aslt w s aslt
4 (16) Unw sex wo force
5 (02) Attempted rape
6 (15) Sex aslt wo inj
7 (18) Verbal thr rape
8 (01) Completed rape
```

The estimated number of all sexual assaults in 2022 in the United States is

```
dataExt |>
  filter(grepl("rape|[Ss]ex", V4529)) |>
  summarize(sum(SERIES_WEIGHT))
```

```
sum(SERIES_WEIGHT)
1                457268
```

Since 2019 we have had no national crime estimates based on police reports.

6 Calculating victimization by demographics

In the remainder of these notes, we will examine relationships between victimization and the respondents' features, like age (V3014), marital status (V3015), and sex (V3018). To make the code more clear, let's give these variables more intelligible names.

```
dataExt <- dataExt |>
  rename(age=V3014, marital=V3015, sex=V3018)
```

Perhaps we are interested in which crimes disproportionately affect men and which disproportionately affect women. Start by tabulating crime type by sex.

```
dataExt |>
  group_by(V4529, sex) |>
  summarize(count=sum(SERIES_WEIGHT)) |>
  print(n=Inf)
```

``summarise()`` has grouped output by 'V4529'. You can override using the ``groups`` argument.

```
# A tibble: 65 x 3
# Groups:   V4529 [33]
  V4529          sex      count
  <fct>      <fct>    <dbl>
1 (01) Completed rape (1) Male    18080.
2 (01) Completed rape (2) Female 109442.
3 (02) Attempted rape (1) Male     2190.
4 (02) Attempted rape (2) Female  65140.
```

5	(03)	Sex aslt w s aslt	(2)	Female	27038.
6	(04)	Sex aslt w m aslt	(1)	Male	677.
7	(04)	Sex aslt w m aslt	(2)	Female	6834.
8	(05)	Rob w inj s aslt	(1)	Male	52000.
9	(05)	Rob w inj s aslt	(2)	Female	35026.
10	(06)	Rob w inj m aslt	(1)	Male	16581.
11	(06)	Rob w inj m aslt	(2)	Female	72039.
12	(07)	Rob wo injury	(1)	Male	102379.
13	(07)	Rob wo injury	(2)	Female	114209.
14	(08)	At rob inj s asl	(1)	Male	16131.
15	(08)	At rob inj s asl	(2)	Female	9065.
16	(09)	At rob inj m asl	(1)	Male	31641.
17	(09)	At rob inj m asl	(2)	Female	16255.
18	(10)	At rob w aslt	(1)	Male	57154.
19	(10)	At rob w aslt	(2)	Female	62125.
20	(11)	Ag aslt w injury	(1)	Male	195301.
21	(11)	Ag aslt w injury	(2)	Female	254541.
22	(12)	At ag aslt w wea	(1)	Male	189557.
23	(12)	At ag aslt w wea	(2)	Female	207692.
24	(13)	Thr aslt w weap	(1)	Male	402951.
25	(13)	Thr aslt w weap	(2)	Female	235001.
26	(14)	Simp aslt w inj	(1)	Male	214975.
27	(14)	Simp aslt w inj	(2)	Female	363093.
28	(15)	Sex aslt wo inj	(1)	Male	22433.
29	(15)	Sex aslt wo inj	(2)	Female	95645.
30	(16)	Unw sex wo force	(1)	Male	7980.
31	(16)	Unw sex wo force	(2)	Female	44446.
32	(17)	Asl wo weap, wo inj	(1)	Male	648373.
33	(17)	Asl wo weap, wo inj	(2)	Female	647384.
34	(18)	Verbal thr rape	(1)	Male	15134.
35	(18)	Verbal thr rape	(2)	Female	26727.
36	(19)	Ver thr sex aslt	(1)	Male	5480.
37	(19)	Ver thr sex aslt	(2)	Female	10021.
38	(20)	Verbal thr aslt	(1)	Male	1021068.
39	(20)	Verbal thr aslt	(2)	Female	957555.
40	(21)	Purse snatching	(1)	Male	1947.
41	(21)	Purse snatching	(2)	Female	33428.
42	(23)	Pocket picking	(1)	Male	92254.
43	(23)	Pocket picking	(2)	Female	32249.
44	(31)	Burg, force ent	(1)	Male	177579.
45	(31)	Burg, force ent	(2)	Female	371623.
46	(32)	Burg, ent wo for	(1)	Male	467653.
47	(32)	Burg, ent wo for	(2)	Female	606561.

48	(33)	Att force entry	(1) Male	140494.
49	(33)	Att force entry	(2) Female	172036.
50	(40)	Motor veh theft	(1) Male	249986.
51	(40)	Motor veh theft	(2) Female	268811.
52	(41)	At mtr veh theft	(1) Male	91918.
53	(41)	At mtr veh theft	(2) Female	101140.
54	(54)	Theft < \$10	(1) Male	255762.
55	(54)	Theft < \$10	(2) Female	367273.
56	(55)	Theft \$10-\$49	(1) Male	820707.
57	(55)	Theft \$10-\$49	(2) Female	914912.
58	(56)	Theft \$50-\$249	(1) Male	1377060.
59	(56)	Theft \$50-\$249	(2) Female	1591759.
60	(57)	Theft \$250+	(1) Male	1649266.
61	(57)	Theft \$250+	(2) Female	1408679.
62	(58)	Theft value NA	(1) Male	809339.
63	(58)	Theft value NA	(2) Female	989933.
64	(59)	Attempted theft	(1) Male	400789.
65	(59)	Attempted theft	(2) Female	348795.

R produces a long narrow table. This format is sometimes useful, particularly when merging data frames. However, in this case having a table with counts for men and women side-by-side would be easier to absorb. `pivot_wider()` will swing the sex column into two side-by-side columns.

```
dataExt |>
  group_by(V4529, sex) |>
  summarize(count=sum(SERIES_WEIGHT)) |>
  pivot_wider(names_from=sex,
              values_from=count,
              values_fill = 0) |> # fill in NAs with 0
  print(n=Inf)
```

``summarise()`` has grouped output by 'V4529'. You can override using the ``.groups`` argument.

```
# A tibble: 33 x 3
# Groups:   V4529 [33]
  V4529                `(1) Male` `(2) Female`
  <fct>                <dbl>         <dbl>
1 (01) Completed rape  18080.         109442.
2 (02) Attempted rape   2190.          65140.
```

3	(03)	Sex aslt w s aslt	0	27038.
4	(04)	Sex aslt w m aslt	677.	6834.
5	(05)	Rob w inj s aslt	52000.	35026.
6	(06)	Rob w inj m aslt	16581.	72039.
7	(07)	Rob wo injury	102379.	114209.
8	(08)	At rob inj s asl	16131.	9065.
9	(09)	At rob inj m asl	31641.	16255.
10	(10)	At rob w aslt	57154.	62125.
11	(11)	Ag aslt w injury	195301.	254541.
12	(12)	At ag aslt w wea	189557.	207692.
13	(13)	Thr aslt w weap	402951.	235001.
14	(14)	Simp aslt w inj	214975.	363093.
15	(15)	Sex aslt wo inj	22433.	95645.
16	(16)	Unw sex wo force	7980.	44446.
17	(17)	Asl wo weap, wo inj	648373.	647384.
18	(18)	Verbal thr rape	15134.	26727.
19	(19)	Ver thr sex aslt	5480.	10021.
20	(20)	Verbal thr aslt	1021068.	957555.
21	(21)	Purse snatching	1947.	33428.
22	(23)	Pocket picking	92254.	32249.
23	(31)	Burg, force ent	177579.	371623.
24	(32)	Burg, ent wo for	467653.	606561.
25	(33)	Att force entry	140494.	172036.
26	(40)	Motor veh theft	249986.	268811.
27	(41)	At mtr veh theft	91918.	101140.
28	(54)	Theft < \$10	255762.	367273.
29	(55)	Theft \$10-\$49	820707.	914912.
30	(56)	Theft \$50-\$249	1377060.	1591759.
31	(57)	Theft \$250+	1649266.	1408679.
32	(58)	Theft value NA	809339.	989933.
33	(59)	Attempted theft	400789.	348795.

Lastly, let's normalize the columns so that they add up to 100%, giving us the distribution of crime types within each sex.

```
dataExt |>
  group_by(V4529, sex) |>
  summarize(count=sum(SERIES_WEIGHT)) |>
  ungroup() |> # without this, percentages are computed within crime type
  pivot_wider(names_from = sex,
               values_from = count,
               values_fill = 0) |>
```

```

rename(male=`(1) Male`, female=`(2) Female`) |>
mutate(male =100*male/sum(male),
       female=100*female/sum(female),
       ratio=female/male) |>
arrange(desc(ratio)) |>
print(n=Inf)

```

`summarise()` has grouped output by 'V4529'. You can override using the
 `groups` argument.

A tibble: 33 x 4

V4529	male	female	ratio
<fct>	<dbl>	<dbl>	<dbl>
1 (03) Sex aslt w s aslt	0	0.256	Inf
2 (02) Attempted rape	0.0229	0.616	26.9
3 (21) Purse snatching	0.0204	0.316	15.5
4 (04) Sex aslt w m aslt	0.00709	0.0647	9.12
5 (01) Completed rape	0.189	1.04	5.47
6 (16) Unw sex wo force	0.0835	0.421	5.04
7 (06) Rob w inj m aslt	0.174	0.682	3.93
8 (15) Sex aslt wo inj	0.235	0.905	3.86
9 (31) Burg, force ent	1.86	3.52	1.89
10 (19) Ver thr sex aslt	0.0574	0.0948	1.65
11 (18) Verbal thr rape	0.158	0.253	1.60
12 (14) Simp aslt w inj	2.25	3.44	1.53
13 (54) Theft < \$10	2.68	3.48	1.30
14 (11) Ag aslt w injury	2.04	2.41	1.18
15 (32) Burg, ent wo for	4.89	5.74	1.17
16 (33) Att force entry	1.47	1.63	1.11
17 (58) Theft value NA	8.47	9.37	1.11
18 (56) Theft \$50-\$249	14.4	15.1	1.05
19 (07) Rob wo injury	1.07	1.08	1.01
20 (55) Theft \$10-\$49	8.59	8.66	1.01
21 (41) At mtr veh theft	0.962	0.957	0.995
22 (12) At ag aslt w wea	1.98	1.97	0.991
23 (10) At rob w aslt	0.598	0.588	0.983
24 (40) Motor veh theft	2.62	2.54	0.972
25 (17) Asl wo weap, wo inj	6.79	6.13	0.903
26 (20) Verbal thr aslt	10.7	9.06	0.848
27 (59) Attempted theft	4.19	3.30	0.787
28 (57) Theft \$250+	17.3	13.3	0.772

29	(05)	Rob w inj s aslt	0.544	0.331	0.609
30	(13)	Thr aslt w weap	4.22	2.22	0.527
31	(08)	At rob inj s asl	0.169	0.0858	0.508
32	(09)	At rob inj m asl	0.331	0.154	0.465
33	(23)	Pocket picking	0.966	0.305	0.316

Sexual assaults disproportionately affect women, while pocket picking and attempted robbery involving assaults disproportionately affect men.

The sequence `group_by()`, `summarize()`, `ungroup()` is so common that there is an alternative way to do the same calculation more compactly with `.by` in `summarize()`. A frequent R error is to use `group_by()`, then forget that the data is still grouped, and continue to do calculations unaware that they are occurring only within groups. The `.by` argument also helps avoid this error.

```
# can reduce the group_by, summarize, ungroup with .by
dataExt |>
  summarize(count=sum(SERIES_WEIGHT),
            .by = c(V4529, sex)) |> # eliminates group/ungroup
  pivot_wider(names_from = sex,
              values_from = count,
              values_fill = 0) |>
  rename(male=`(1) Male`, female=`(2) Female`) |>
  mutate(male = 100*male/sum(male),
         female=100*female/sum(female),
         ratio=female/male) |>
  arrange(desc(ratio)) |>
  print(n=Inf)
```

```
# A tibble: 33 x 4
  V4529      male female  ratio
<fct>    <dbl>   <dbl>   <dbl>
1 (03) Sex aslt w s aslt      0    0.256  Inf
2 (02) Attempted rape    0.0229  0.616  26.9
3 (21) Purse snatching  0.0204  0.316  15.5
4 (04) Sex aslt w m aslt  0.00709 0.0647   9.12
5 (01) Completed rape    0.189    1.04   5.47
6 (16) Unw sex wo force  0.0835  0.421   5.04
7 (06) Rob w inj m aslt   0.174    0.682   3.93
8 (15) Sex aslt wo inj    0.235    0.905   3.86
9 (31) Burg, force ent    1.86    3.52   1.89
10 (19) Ver thr sex aslt  0.0574  0.0948   1.65
```

11	(18)	Verbal thr rape	0.158	0.253	1.60
12	(14)	Simp aslt w inj	2.25	3.44	1.53
13	(54)	Theft < \$10	2.68	3.48	1.30
14	(11)	Ag aslt w injury	2.04	2.41	1.18
15	(32)	Burg, ent wo for	4.89	5.74	1.17
16	(33)	Att force entry	1.47	1.63	1.11
17	(58)	Theft value NA	8.47	9.37	1.11
18	(56)	Theft \$50-\$249	14.4	15.1	1.05
19	(07)	Rob wo injury	1.07	1.08	1.01
20	(55)	Theft \$10-\$49	8.59	8.66	1.01
21	(41)	At mtr veh theft	0.962	0.957	0.995
22	(12)	At ag aslt w wea	1.98	1.97	0.991
23	(10)	At rob w aslt	0.598	0.588	0.983
24	(40)	Motor veh theft	2.62	2.54	0.972
25	(17)	Asl wo weap, wo inj	6.79	6.13	0.903
26	(20)	Verbal thr aslt	10.7	9.06	0.848
27	(59)	Attempted theft	4.19	3.30	0.787
28	(57)	Theft \$250+	17.3	13.3	0.772
29	(05)	Rob w inj s aslt	0.544	0.331	0.609
30	(13)	Thr aslt w weap	4.22	2.22	0.527
31	(08)	At rob inj s asl	0.169	0.0858	0.508
32	(09)	At rob inj m asl	0.331	0.154	0.465
33	(23)	Pocket picking	0.966	0.305	0.316

We can do a similar calculation by age. First, let's discretize age into some fixed age bins. Then, we can repeat the same calculation to learn about victimization differences by age. I have sorted the results by the 18-24 age category, but you can change it to your age category if you wish.

```
# can reduce the group_by, summarize, ungroup with .by
dataExt |>
  mutate(ageGroup =
    cut(age,
        breaks = c(0, 17, 24, 34, 49, 64, Inf),
        labels = c("12-17", "18-24", "25-34",
                   "35-49", "50-64", "65+"))) |>
  summarize(count=sum(SERIES_WEIGHT),
    .by = c(V4529, ageGroup)) |>
  pivot_wider(names_from = ageGroup,
    values_from = count,
    values_fill = 0,
    names_sort = TRUE) |> # keep age groups ordered
```

```
# apply the same function to every column, except V4529
mutate(across(-V4529, function(x) 100*x/sum(x))) |>
arrange(desc(`18-24`)) |> # you can change to your age group
print(n=Inf)
```

A tibble: 33 x 7

V4529	`12-17`	`18-24`	`25-34`	`35-49`	`50-64`	`65+`
<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 (56) Theft \$50-\$249	9.31	13.0	16.2	16.1	14.9	12.9
2 (57) Theft \$250+	4.04	11.8	16.1	17.7	16.2	14.3
3 (20) Verbal thr aslt	14.2	9.81	10.1	8.65	11.1	7.99
4 (55) Theft \$10-\$49	4.14	8.81	9.11	8.16	7.48	12.1
5 (58) Theft value NA	4.70	7.41	8.63	8.71	10.6	10.1
6 (17) Asl wo weap, wo inj	23.5	6.29	6.27	7.01	3.88	4.09
7 (13) Thr aslt w weap	3.63	4.74	3.33	3.13	3.01	1.50
8 (32) Burg, ent wo for	0.743	4.25	3.61	5.00	6.36	9.99
9 (12) At ag aslt w wea	4.90	3.56	1.38	1.91	1.73	0.924
10 (11) Ag aslt w injury	3.39	3.52	2.29	2.04	2.33	0.648
11 (31) Burg, force ent	0.319	3.44	1.63	3.23	3.00	3.33
12 (14) Simp aslt w inj	9.17	3.42	3.22	2.11	2.53	1.59
13 (59) Attempted theft	1.12	2.93	3.86	3.71	3.91	4.94
14 (40) Motor veh theft	0.657	2.62	2.39	2.78	2.75	2.85
15 (54) Theft < \$10	4.11	2.45	3.74	2.81	2.69	3.54
16 (15) Sex aslt wo inj	2.30	1.76	0.432	0.273	0.409	0
17 (01) Completed rape	0.202	1.60	1.04	0.516	0.0606	0.340
18 (07) Rob wo injury	1.77	1.36	0.992	0.767	1.07	1.31
19 (06) Rob w inj m aslt	0.949	1.26	0.878	0.0208	0.0447	0.197
20 (02) Attempted rape	0.239	1.10	0.113	0.377	0.214	0.110
21 (21) Purse snatching	0	1.04	0	0	0.119	0.126
22 (33) Att force entry	0	0.695	1.80	1.56	1.47	2.67
23 (10) At rob w aslt	0.595	0.642	0.511	0.458	0.751	0.681
24 (23) Pocket picking	1.66	0.432	0.438	0.489	0.543	1.14
25 (41) At mtr veh theft	0	0.426	0.880	1.25	1.26	0.883
26 (08) At rob inj s asl	0.979	0.397	0.0819	0.0448	0	0
27 (16) Unw sex wo force	2.17	0.387	0.120	0	0.246	0.245
28 (18) Verbal thr rape	0.128	0.303	0.157	0.185	0.348	0.0321
29 (05) Rob w inj s aslt	0	0.187	0.399	0.258	0.846	0.525
30 (03) Sex aslt w s aslt	0	0.172	0.0642	0	0	0.771
31 (09) At rob inj m asl	0.363	0.133	0.120	0.566	0.0915	0.103
32 (19) Ver thr sex aslt	0.293	0	0.115	0.0918	0.0167	0.0857
33 (04) Sex aslt w m aslt	0.418	0	0	0.0421	0.0345	0

Since we have made many changes to the dataset, I find it useful to save the final version. This way I can simply `load()` the data again later and know that it already has all the edits and changes that I have made.

```
save(dataExt, file="NCVS2022.RData", compress=TRUE)
```

Fisher, Bonnie S., and Rachel L. Gross. 2025. "The Evolution of the Measurement of Rape and Sexual Assault over 50 Years: Milestones, Definitions, Operationalizations, and Classifications." *Journal of Contemporary Criminal Justice* 41 (1): 166–95. <https://doi.org/10.1177/10439862241290352>.