Introduction to NIBRS

Greg Ridgeway

2025-09-21

Table of contents

1 National Incident-Based Reporting System

The FBI established the Uniform Crime Reporting (UCR) Program in 1930 to generate reliable crime statistics for law enforcement administration, operation, and management. It has been the primary source of crime data in the United States for decades.

The UCR's Summary Reporting System (SRS) is the best-known component of the UCR. About 18,000 law enforcement agencies, including municipal police departments, sheriff's departments, campus police, transit police, park police, and many other agencies, reported monthly counts of Part 1 crimes (murder and non-negligent manslaughter, forcible rape, robbery, aggravated assault, burglary, larceny theft, motor vehicle theft, and arson), often called "index crimes." The SRS also tracked monthly counts of Part II crimes, such as simple assault, fraud, vandalism, drug offenses, and driving under the influence. Even though reporting to the SRS was voluntary, almost all law enforcement agencies reported their data, offering fairly comprehensive coverage of crimes reported to the police in the United States.

While the UCR SRS has provided valuable crime data for many years, it has several notable limitations. The SRS focused on a limited number of crimes, potentially overlooking important details and emerging trends. With SRS data researchers cannot separate out trends in shootings, identity theft, and cybercrimes. Shootings, for example, are lumped together in an aggravated assault category that includes a range of serious assaults like striking someone with a beer bottle. The SRS also operated under the "Hierarchy Rule," reporting only the most severe offense in a multi-offense incident. Lastly, the SRS only collected aggregate counts, meaning that detailed information about the context of the crimes, such as the characteristics of the victims and offenders, the extent of property loss or damage, the time, place, and context of the crime, and the relationships between these crime features, is lost.

The FBI introduced the National Incident-Based Reporting System (NIBRS) in the 1980s. NIBRS aimed to address the shortcomings of the SRS by capturing incident-level data and a comprehensive description of what happened in each incident. Unlike the SRS, NIBRS collects data on each individual crime incident, capturing detailed information about the offenses, victims, offenders, property, and arrestees. NIBRS records data on 52 "Group A" offenses and 10 "Group B" offenses, covering a broader spectrum of criminal activity. It did away with the Hierarchy Rule and collects data on all offenses within a single incident, providing a fuller picture of criminal activity. Because NIBRS is incident-based, we have full access to the multivariate relationships between features of crime incidents.

On January 1, 2021, the FBI officially retired the SRS marking a significant transition towards the exclusive use of the National Incident-Based Reporting System for crime data collection and reporting. Although the transition has been planned for almost a decade, many law enforcement agencies are yet to transition their information systems to report to NIBRS. As of this writing, four of the nation's largest states, California, New York, Pennsylvania, and Florida, essentially do not participate in NIBRS. You can find a map describing NIBRS coverage at the Bureau of Justice Statistics. The largest law enforcement agencies in those states regularly post crime data to their local open data portals, but you will not find those data in NIBRS yet. The transition to NIBRS also makes the study of long-term national crime trends challenging. Any study that spans the SRS-NIBRS transition will have to grapple with inconsistencies and gaps in data as agencies adapt to the new system.

The FBI accepts data from law enforcement agencies through March of the following year. That is, the FBI accepted NIBRS data for 2023 through March 2024. Some crimes committed in December 2023, for example, may be solved or result in an arrest in April 2024. Such a case would not get marked as "cleared" in the NIBRS data since the clearance came after the NIBRS 2023 submission end date. Crimes in January 2023 may seem to be solved at a higher rate than crimes in December 2023 because of the March 2024 censoring. This is a feature of the data that simply requires care. Lastly, like the SRS, NIBRS only has data on those crimes reported to the police and reporting rates can vary greatly by type of crime.

In this document, you will walk through running the nibrs.R script, which processes NIBRS data from 2023. The steps will cover reading the dataset, segmenting data by type (e.g., offenders, victims), and performing some exploration of the NIBRS data.

2 Acquiring the data

The complete NIBRS data are available from the FBI's Crime Data Explorer downloads page under the Master File Downloads section heading. from the dropdown menu select "National Incident-Based Reporting System (NIBRS)". The compressed data file, `nibrs-2023.zip`, is over 500 Mb so downloading will take some time. It has over 73 million rows of data. Unzip `nibrs-2023.zip`. It is very big... about 6 Gb. I then run `gzip 2023_NIBRS_NATIONAL_MASTER_FILE.TXT` from the Terminal pane in RStudio (not the Console pane, the Terminal pane is typically one tab to the right of the Console pane).

Let's start by loading a few libraries that we will need and peeking at a few lines of the data.

```r
library(readxl)
library(dplyr)
library(tidyr)
library(readr)
library(stringr)
library(lubridate)

scan("2023_NIBRS_NATIONAL_MASTER_FILE.txt.gz",
     nlines=5, what="", sep="\n")
```

```
[1] "BH50AK00101000000000000000                 ANCHORAGE                       AK1C941Y
[2] "BH50AK00102000000000000000         20210101FAIRBANKS                       AK4 941Y
```

```
[3] "0150AK001020023000003    20230101 11010010100    N
[4] "0250AK001020023000003    2023010123DCN  04                 88"
[5] "0350AK001020023000003    20230101775000000150
```

As you can see the data are not pretty. These data are in "fixed-width format". Rather than separate each column of data with a delimiter (like a comma or tab), all of the data fields are placed side by side. This is a legacy format that has some benefits. First, there is no need to store delimiters. This is less important these days with cheap data storage, but storing delimiters in a dataset this size requires about 1 Gb... just to store a bunch of commas. Instead, the data come with a separate file that describes which columns of text correspond to different columns of data. The following table shows the first five rows of the NIBRS offense segment in the included `NIBRS Records Description updated.xlsx` file.

Positions	Field Length and Type	Field Name
1-2	A2	SEGMENT LEVEL
3-4	N2	NUMERIC STATE CODE
5-13	A9	ORIGINATING AGENCY IDENTIFIER (ORI)
14-25	A12	INCIDENT NUMBER
26-33	A8	INCIDENT DATE
34-36	A3	UCR OFFENSE CODE
...	...	...

The table tells us that for offense segments, the first two characters represent the segment level, characters 3 and 4 capture the state numeric code, characters 5-13 capture the ORI (a unique identifier for a law enforcement agency), and so on. The second column describes the data type, (A)lphanumeric or (N)umeric, and the width (number of characters) of the data field. Let's start by discussing the segment level. The NIBRS data is more complex than an already complex fixed-width format data file. The NIBRS data involve several data tables, separate tables for offenses, victims, offenders, property, and arrests. The data file we just loaded interleaves all of these tables. Note that the fourth row of the data we scanned in starts with "02" that signals that the rest of that row describes an offense and its format will align with the formatting table shown here. Since characters 3-4 are "50" we know from the format table that this number represents the state with code "50," which turns out to be Alaska. The next nine characters (AK0010200) is the ORI code for the law enforcement agency that reported the offense, which turns out to be the Fairbanks Police Department. If we look further to the right in that fourth row of data to characters 26-33 we get "20230101," which is the data of the criminal incident, January 1, 2023. And the three characters after the date give the offense code "23D," which is the code for theft from a building.

This interpretation only works for lines of data with the first two characters equal to "02". The others are

Segment code	Segment type
BH	batch header
01	administrative
02	offenses
03	property
04	victims
05	offender
06	arrestee
07	Group B arrests
W1	Incomplete admin
W3	Incomplete property
W6	Incomplete arrest

We will read each of these in and explore what they contain. The W1, W2, and W3 are "window segments" and represent a partial reporting. These are relatively rare records and mostly relate to arrests or recovered property related to offenses that do not appear in the 02 segment, most likely because the agency transitioned to NIBRS between the offense and the recovery/arrest.

3 Reading in the data

Here's the strategy that we will use to read in the data. The data might be too big to read in all at once and still have computer memory available to clean and organize the data. Instead of reading all of it in, we are going to read in one million rows at a time. We will then examine the first two characters of each row and split rows of data by those first two characters. In this way, all the offense records will be together, and all the victim records will be together, and so on. Then we will write out all the records into segment specific files. Our computers will then have a separate file for each of the NIBRS segments.

First, I set up `infile` to connect to the large NIBRS data file, opening it in (r)ead mode. Next, I set up eight files, one for each of the segments, creating them in (w)rite mode to be gzip'd (compressed) as they are created. Then I set up a while-loop to continue to read one million lines at a time as long as there are more rows of data to read in. Within the loop I use `split()` to separate the lines of data I have read in based on the first two characters of the line. I combine the few window segments with the similar complete records (e.g. window segment on property goes with the rest of the property data). Lastly, inside the loop I write out the new batch of data, appending them to existing data from previous iterations of the while-loop. Closing all the files makes sure that all the read and write buffers get flushed to the data files and the connections are closed.

```r
# set the file we want to 'r'ead
infile  <- file("2023_NIBRS_NATIONAL_MASTER_FILE.txt.gz",'r')
# create the files we are going to 'w'rite
outfile <- list("01"=gzfile("2023-01.txt.gz", 'w'),
                "02"=gzfile("2023-02.txt.gz", 'w'),
                "03"=gzfile("2023-03.txt.gz", 'w'),
                "04"=gzfile("2023-04.txt.gz", 'w'),
                "05"=gzfile("2023-05.txt.gz", 'w'),
                "06"=gzfile("2023-06.txt.gz", 'w'),
                "07"=gzfile("2023-07.txt.gz", 'w'),
                "BH"=gzfile("2023-BH.txt.gz", 'w'))

# read in 1,000,000 lines at a time
cLines <- 0
while ((length(a <- readLines(infile, n=1000000)) >  0))
{
  # split up what we read based on first two characters BH, 01, 02, ...
  b <- split(a, substring(a,1,2))

  # combine window segments with the associated segments
  b[["01"]] <- c(b[["01"]], b[["W1"]]) # administrative
  b[["03"]] <- c(b[["03"]], b[["W3"]]) # property
  b[["06"]] <- c(b[["06"]], b[["W6"]]) # arrestees
  b[c("W1","W3","W6")] <- NULL # drop the W1, W3, W6

  # write each segment to its own file
  for(iSegment in names(b))
  {
    writeLines(b[[iSegment]], con=outfile[[iSegment]])
  }
}
close(infile)
for(iSegment in 1:8)
{
  close(outfile[[iSegment]])
}
```

Now with the segments separated into their own files, we can read them each in individually.

The Excel file `NIBRS Records Description updated.xlsx` contains all the information about which columns in which segments contain which data field. It is probably a good idea to open the file in Excel and explore what is in there and how the file is structured. We will start by reading in the second sheet called "INCIDENT RECORD" using `read_excel()` from the readxl library. Note that range is set to just capture the part of the sheet with the information that we need. So, the first row in fmtXL will be the row with the header: Data Field Number, Position, Type/Length, Description.

```
fmtXL <- read_excel("NIBRS Records Description updated.xlsx",
                    sheet = "INCIDENT RECORD",
                    range = "A5:D819") |>
   rename(DataField=`Data Field Number`,
          TypeLength=`Type/ Length`)
```

3.1 Administrative segment

We will start by reading in segment 01, the administrative segment, where the records description lies between rows 3 and 100 in fmtXL. I'm going to use grep() to find out where the description of the next segment (LEVEL 02, the offense segment) starts so I know where the end of the administrative segment is. I am also going to drop those rows where Position is missing and where Position=="59-88", which is redundant with other record descriptions which provide more details about the contents between characters 59 and 88 in administrative records.

```
# select only the rows for admin segment
i <- grep('LEVEL "02"', fmtXL$DataField)
fmt <- fmtXL |>
  slice(3:(i-3)) |>
  filter(!is.na(Position) &
         !(Position %in% c("59-88"))) # up to 10 UCR codes
fmt
```

```
# A tibble: 25 x 4
   DataField Position TypeLength Description
   <chr>     <chr>    <chr>      <chr>
 1 <NA>      1-2      A2         "SEGMENT LEVEL - Designates this as the Admini~
 2 <NA>      3-4      N2         "NUMERIC STATE CODE - This is a two-digit valu~
 3 1         5-13     A9         "ORIGINATING AGENCY IDENTIFIER (ORI) - This id~
 4 2         14-25    A12        "INCIDENT NUMBER - This is a unique case numbe~
 5 3         26-33    A8         "INCIDENT DATE - This is the date that the inc~
 6 <NA>      34       A1         "REPORT DATE INDICATOR - If value is \"R\" = R~
 7 <NA>      35-36    A2         "INCIDENT DATE HOUR - This is the military tim~
 8 <NA>      37-38    N2         "TOTAL OFFENSE SEGMENTS - This is the number o~
 9 <NA>      39-41    N3         "TOTAL VICTIM SEGMENTS - This is the number of~
10 <NA>      42-43    N2         "TOTAL OFFENDER SEGMENTS - This is the number ~
# i 15 more rows
```

`fmt` now contains everything we need to know about how to interpret the administrative data. We are going to use `read_fwf()` from the `readr` package to read in the fixed-width formatted administrative segment (R also has a `read.fwf()` function, but it is extremely slow with large files). `read_fwf()` needs to know how wide each column is and whether it is of type character (c) or of type numeric (n). The `TypeLength` column in `fmt` has this information. If the first character is an A (alphanumeric) then I will set `fmtType` to "c", otherwise I will set it to "n". Then the rest of the `TypeLength` field I will store in `fmtWidth`. Lastly, I will need to create some column names for the dataset and I see that the part of the `Description` column before the hyphen has a suitable column name. I'll use `strsplit()` to separate the `Description` at the " - " and keep only the first part for the column names. `make.names()` converts the names to syntactically valid R variable names, replacing invalid characters like spaces and # with a period instead.

```r
fmtType <- fmt$TypeLength |>
   substring(1,1) |>            # get first character
   case_match("A"~"c","N"~"n")  # convert A->c, N->n
fmtWidth <- fmt$TypeLength |>
   substring(2) |>   # get everything after the first character
   as.numeric()      # turn text to a number
fmtNames <- fmt$Description |>
   strsplit(split = " - ", fixed = TRUE) |>
   sapply(head, n=1) |>
   make.names() |>
   gsub("\\.+(\\.|$)", "\\1", x=_) # turn .. into one . or delete trailing .
```

Time to read in the administrative segment.

```r
# read in the data from fixed width format
#    R also has a read.fwf() function but it is very slow
nibrsAdmin01 <-
   read_fwf("2023-01.txt.gz",
            col_positions = fwf_widths(fmtWidth),
            col_type=paste(fmtType, collapse="")) |>
   rename_with(~fmtNames) |>
   rename(ORI     = ORIGINATING.AGENCY.IDENTIFIER.ORI,
          SEGMENT = SEGMENT.LEVEL,
          STATE   = NUMERIC.STATE.CODE)
```

Let's take a look at the first three rows.

```r
nibrsAdmin01 |> print(n=3, width=Inf)
```

```
# A tibble: 12,351,743 x 25
  SEGMENT STATE ORI       INCIDENT.NUMBER INCIDENT.DATE REPORT.DATE.INDICATOR
  <chr>   <dbl> <chr>     <chr>           <chr>         <chr>
1 01         50 AK0010200 23000003        20230101      <NA>
2 01         50 AK0010200 23000005        20230101      <NA>
3 01         50 AK0010200 23000011        20230102      <NA>
  INCIDENT.DATE.HOUR TOTAL.OFFENSE.SEGMENTS TOTAL.VICTIM.SEGMENTS
  <chr>                               <dbl>                 <dbl>
1 11                                      1                     1
2 15                                      1                     1
3 16                                      1                     1
  TOTAL.OFFENDER.SEGMENTS TOTAL.ARRESTEE.SEGMENTS CITY.SUBMISSION
                    <dbl>                   <dbl> <chr>
1                       1                       0 <NA>
2                       1                       0 <NA>
3                       2                       0 <NA>
  CLEARED.EXCEPTIONALLY EXCEPTIONAL.CLEARANCE.DATE OFFENSE.CODE.1 OFFENSE.CODE.2
  <chr>                 <chr>                      <chr>          <chr>
1 N                     <NA>                       <NA>           <NA>
2 N                     <NA>                       <NA>           <NA>
3 N                     <NA>                       <NA>           <NA>
  OFFENSE.CODE.3 OFFENSE.CODE.4 OFFENSE.CODE.5 OFFENSE.CODE.6 OFFENSE.CODE.7
  <chr>          <chr>          <chr>          <chr>          <chr>
1 <NA>           <NA>           <NA>           <NA>           <NA>
2 <NA>           <NA>           <NA>           <NA>           <NA>
3 <NA>           <NA>           <NA>           <NA>           <NA>
  OFFENSE.CODE.8 OFFENSE.CODE.9 OFFENSE.CODE.10 Cargo.Theft.Indicator
  <chr>          <chr>          <chr>           <chr>
1 <NA>           <NA>           <NA>            N
2 <NA>           <NA>           <NA>            N
3 <NA>           <NA>           <NA>            N
# i 12,351,740 more rows
```

The first row describes one incident, occurring in Alaska (STATE=50), reported to the Fairbanks Police Department (ORI=AK0010200), that occurred on January 1, 2023 at around 11am. The incident number, 23000003, is only unique within an agency. There may be other incidents that other law enforcement agencies reported that have the same incident number. Whenever linking this record to learn more about the offense, victims, property, and arrests, be sure to join on *both* the ORI *and* the incident number. The first row also tells us that this incident involved one particular offense, involving one victim and one offender, with no arrests (at least through the cutoff date of March 15, 2024).

3.2 Offense segment

Let's move on to loading the offense data. NIBRS reports all "Group A offenses" in the offense table. Group A offenses are mostly crimes against a person or property. NIBRS includes crime types based on

- seriousness of offense
- frequency of its occurrence
- prevalence nationwide
- probability the offense comes to police attention
- law enforcement is the best channel for collecting data
- burden placed on law enforcement to collect data
- validity and usefulness of the collected data
- legitimate general interest in offense

A complete list of offenses that fall into Group A can be found at the Bureau of Justice Statistics website. Generally, Group A includes arson, assault, burglary, counterfeiting, vandalism, drugs, extortion, fraud, gambling, homicide, human trafficking, kidnapping, theft, motor vehicle theft, obscene material, prostitution, robbery, sex offenses, and weapons offenses. You will not find reports of Group B offenses in the offense segment. Group B offenses include writing bad checks, curfew violations, loitering, disorderly conduct, drunk driving, family offenses, peeping tom, trespassing, and other non-violent offenses. These crimes only get included in NIBRS arrest segment if there is an arrest. Reports from the public about Group B offenses do not make it into the offense segment.

As with the administrative segment, we will extract from the Excel records description the offense segment's format.

```r
# select only the rows for offense segment,
#   everything between LEVEL 02 and LEVEL 03
i <- grep('LEVEL "02"', fmtXL$DataField)
j <- grep('LEVEL "03"', fmtXL$DataField)
fmt <- fmtXL |>
  slice((i+1):(j-3)) |>
  filter(!is.na(Position) &
         !(Position %in% c("38-40","46-48","49-57")))
fmt
```

```
# A tibble: 21 x 4
   DataField Position TypeLength Description
   <chr>      <chr>    <chr>      <chr>
 1 <NA>       1-2      A2         "SEGMENT LEVEL - Designates this as the Offens~
 2 <NA>       3-4      N2         "NUMERIC STATE CODE - This is a two-digit valu~
 3 1          5-13     A9         "ORIGINATING AGENCY IDENTIFIER (ORI) - This id~
 4 2          14-25    A12        "INCIDENT NUMBER - This is a unique case numbe~
 5 3          26-33    A8         "INCIDENT DATE - This is the same date value f~
 6 6          34-36    A3         "UCR OFFENSE CODE"
 7 7          37       A1         "OFFENSE ATTEMPTED/COMPLETED - This indicates ~
 8 8          38       A1         "SUSPECTED OF USING #1"
 9 8          39       A1         "SUSPECTED OF USING #2"
10 8          40       A1         "SUSPECTED OF USING #3"
# i 11 more rows
```

This `fmt` is similar in structure to the one we saw for the administrative segment, just with a different set of features, ones more specific to the offense, like the UCR code of the offense and whether it was attempted or completed. Let's use `read_fwf()` to read in the offense data.

```
fmtType <- fmt$TypeLength |>
   substring(1,1) |>
   case_match("A"~"c","N"~"n")
fmtWidth <- fmt$TypeLength |>
   substring(2) |>
   as.numeric()
fmtNames <- fmt$Description |>
   strsplit(split = " - ", fixed = TRUE) |>
   sapply(head, n=1) |>
   make.names() |>
   gsub("\\.+(\\.|$)", "\\1", x=_)

nibrsCrm02 <- read_fwf("2023-02.txt.gz",
                       col_positions = fwf_widths(fmtWidth),
                       col_type=paste(fmtType,collapse="")) |>
   rename_with(~fmtNames) |>
   rename(ORI     = ORIGINATING.AGENCY.IDENTIFIER.ORI,
          SEGMENT = SEGMENT.LEVEL,
          STATE   = NUMERIC.STATE.CODE)
```

Let's pull up the offense record for the Fairbanks, Alaska incident 2300003 that was in the first row of the administrative segment.

```
nibrsCrm02 |> filter(INCIDENT.NUMBER=="23000003") |> print(n=3, width=80)
```

```
# A tibble: 91 x 21
  SEGMENT STATE ORI       INCIDENT.NUMBER INCIDENT.DATE UCR.OFFENSE.CODE
  <chr>   <dbl> <chr>     <chr>           <chr>         <chr>
1 02         50 AK0010200 23000003        20230101      23D
2 02         50 AK0010900 23000003        20230101      290
3 02          2 AZ0071300 23000003        20230101      520
# i 88 more rows
# i 15 more variables: OFFENSE.ATTEMPTED.COMPLETED <chr>,
#   SUSPECTED.OF.USING.1 <chr>, SUSPECTED.OF.USING.2 <chr>,
#   SUSPECTED.OF.USING.3 <chr>, LOCATION.TYPE <chr>,
#   NUMBER.OF.PREMISES.ENTERED <chr>, METHOD.OF.ENTRY <chr>,
#   CRIMINAL.ACTIVITY.1.GANG.INFORMATION.1 <chr>,
#   CRIMINAL.ACTIVITY.2.GANG.INFORMATION.2 <chr>, ...
```

Note here that I only filtered on the incident number. What we got back were all incidents labeled as incident number 23000003, including those from other police departments. Remember that incident numbers are only unique within a law enforcement agency. So, we need to also specifically request the incident from Fairbanks.

```
nibrsCrm02 |>
   filter(ORI=="AK0010200" & INCIDENT.NUMBER=="23000003") |>
   print(width=Inf)
```

```
# A tibble: 1 x 21
  SEGMENT STATE ORI       INCIDENT.NUMBER INCIDENT.DATE UCR.OFFENSE.CODE
  <chr>   <dbl> <chr>     <chr>           <chr>         <chr>
1 02         50 AK0010200 23000003        20230101      23D
  OFFENSE.ATTEMPTED.COMPLETED SUSPECTED.OF.USING.1 SUSPECTED.OF.USING.2
  <chr>                       <chr>                <chr>
1 C                           N                    <NA>
  SUSPECTED.OF.USING.3 LOCATION.TYPE NUMBER.OF.PREMISES.ENTERED METHOD.OF.ENTRY
  <chr>                <chr>         <chr>                      <chr>
1 <NA>                 04            <NA>                       <NA>
  CRIMINAL.ACTIVITY.1.GANG.INFORMATION.1 CRIMINAL.ACTIVITY.2.GANG.INFORMATION.2
  <chr>                                  <chr>
1 <NA>                                   <NA>
  CRIMINAL.ACTIVITY.3 WEAPON.FORCE.1 AUTOMATIC.INDICATOR.1 WEAPON.FORCE.2
  <chr>               <chr>          <chr>                 <chr>
1 <NA>                <NA>           <NA>                  <NA>
```

```
  WEAPON.FORCE.3 BIAS.MOTIVATION
  <chr>          <chr>
1 <NA>           88
```

This offense record describes a UCR code 23D, which is theft from a building. `OFFENSE.ATTEMPTED.COMPLETED` is `C` meaning it was a completed crime (not attempted). During the offense the offender was suspected of using narcotics (`SUSPECTED.OF.USING.1` marked as `N`). Location type is `04`, which is a Church/Synagogue/Temple.

You have noticed by now that NIBRS has a lot of codes, for states, offenses, location types, and there are many more. It might be handy to have a lookup table so we can join the codes to better English descriptions. All the information is in that Excel records description. Let's write some code to create a UCR lookup table.

```r
i <- which(fmtXL$Description=="720 - Animal Cruelty Offenses - Animal Cruelty")
j <- which(fmtXL$Description=="90Z - All Other Offenses - All Other Offenses")

OffenseLookup <-
   strsplit(fmtXL$Description[i:j], " - ") |>
   do.call(rbind, args=_) |> # stack all the rows on top of each other
   data.frame() |>
   rename(UCRCode=X1, CrimeCat=X2, Crime=X3) |>
   filter(UCRCode!="Group B Offenses")
head(OffenseLookup)
```

```
  UCRCode                CrimeCat              Crime
1     720 Animal Cruelty Offenses     Animal Cruelty
2     200                   Arson              Arson
3     13A        Assault Offenses Aggravated Assault
4     13B        Assault Offenses     Simple Assault
5     13C        Assault Offenses       Intimidation
6     510                 Bribery            Bribery
```

Now for each UCR offense code we can look up its general crime category (`CrimeCat`) and its specific crime type (`Crime`). You can adapt this code to create other lookup tables for other codes, like a state code lookup table.

```r
i <- which(fmtXL$Description=="50  = AK -Alaska")
j <- which(fmtXL$Description=="49  =  WY -  Wyoming")

StateLookup <-
   fmtXL$Description[i:j] |>
   data.frame() |>
   rename(temp=fmtXL.Description.i.j.) |>
   mutate(StateCode=as.numeric(substring(temp,1,2)),
          State=gsub(".*([A-Z][A-Z]).*","\\1", temp)) |>
   select(-temp) |>
   arrange(StateCode)
head(StateLookup)
```

```
  StateCode State
1         1    AL
2         2    AZ
3         3    AR
4         4    CA
5         5    CO
6         6    CT
```

3.2.1 Exercises

1. How many Group A offenses were reported to the Philadelphia Police Department (ORI PAPEP0000)?
2. What is the most commonly reported crime in Philadelphia? Use the OffenseLookup to replace the UCR codes with crime descriptions.

3.3 Property segment

Moving on to the NIBRS property segment, we repeat the same steps as we did before, adapting to get the records description for property reports.

```
# select only the rows for property segment
i <- grep('LEVEL "03"', fmtXL$DataField)
j <- grep('LEVEL "04"', fmtXL$DataField)

fmt <- fmtXL |>
  slice((i+2):(j-3)) |>
  filter(!is.na(Position) &
         !(Position %in% c("14-22","20-22","58-102","58-72","103-132")))

fmtType <- fmt$TypeLength |>
   substring(1,1) |>
   case_match("A"~"c","N"~"n")
```

```
fmtWidth <- fmt$TypeLength |>
   substring(2) |>
   as.numeric()
fmtNames <- fmt$Description |>
   strsplit(split = " - ", fixed = TRUE) |>
   sapply(head, n=1) |>
   make.names() |>
   gsub("\\.+(\\.|$)", "\\1", x=_)
```

There are two columns called `ESTIMATED.QUANTITY`. The second one represents thousandths of the first column. So, I'll rename the second one to have a unique name.

```
fmtNames
```

```
 [1] "SEGMENT.LEVEL"                      "NUMERIC.STATE.CODE"
 [3] "ORIGINATING.AGENCY.IDENTIFIER.ORI"  "INCIDENT.NUMBER"
 [5] "INCIDENT.DATE"                      "TYPE.PROPERTY.LOSS.ETC"
 [7] "PROPERTY.DESCRIPTION"               "VALUE.OF.PROPERTY"
 [9] "DATE.RECOVERED"                     "NUMBER.OF.STOLEN.MOTOR.VEHICLES"
[11] "NUMBER.OF.RECOVERED.MOTOR.VEHICLES" "SUSPECTED.DRUG.TYPE"
[13] "ESTIMATED.QUANTITY"                 "ESTIMATED.QUANTITY"
[15] "TYPE.MEASUREMENT"                   "DRUG.INVOLVEMENT.2"
[17] "DRUG.INVOLVEMENT.3"                 "WINDOW.UCR.OFFENSE.CODE.1"
[19] "WINDOW.UCR.OFFENSE.CODE.2"          "WINDOW.UCR.OFFENSE.CODE.3"
[21] "WINDOW.UCR.OFFENSE.CODE.4"          "WINDOW.UCR.OFFENSE.CODE.5"
[23] "WINDOW.UCR.OFFENSE.CODE.6"          "WINDOW.UCR.OFFENSE.CODE.7"
[25] "WINDOW.UCR.OFFENSE.CODE.8"          "WINDOW.UCR.OFFENSE.CODE.9"
[27] "WINDOW.UCR.OFFENSE.CODE.10"
```

```
fmtNames[which(fmtNames=="ESTIMATED.QUANTITY")[2]] <-
   "ESTIMATED.QUANTITY.1000THS"
```

```
nibrsProp03 <- read_fwf("2023-03.txt.gz",
                         col_positions = fwf_widths(fmtWidth),
                         col_type=paste(fmtType,collapse="")) |>
   rename_with(~fmtNames) |>
   mutate(VALUE.OF.PROPERTY=as.numeric(VALUE.OF.PROPERTY)) |>
   rename(ORI     = ORIGINATING.AGENCY.IDENTIFIER.ORI,
          SEGMENT = SEGMENT.LEVEL,
          STATE   = NUMERIC.STATE.CODE)
```

Let's pull up the property record for the Fairbanks, Alaska incident 2300003 that we previously examined.

```r
nibrsProp03 |>
   filter(ORI=="AK0010200" & INCIDENT.NUMBER=="23000003") |>
   print(width=360)
```

```
# A tibble: 1 x 27
  SEGMENT STATE ORI       INCIDENT.NUMBER INCIDENT.DATE TYPE.PROPERTY.LOSS.ETC
  <chr>   <chr> <chr>     <chr>           <chr>         <chr>
1 03      50    AK0010200 23000003        20230101      7
  PROPERTY.DESCRIPTION VALUE.OF.PROPERTY DATE.RECOVERED NUMBER.OF.STOLEN.MOTOR~1
  <chr>                            <dbl> <chr>          <chr>
1 75                                 150 <NA>           <NA>
  NUMBER.OF.RECOVERED.MOTOR.VEHICLES SUSPECTED.DRUG.TYPE ESTIMATED.QUANTITY
  <chr>                              <chr>                <chr>
1 <NA>                               <NA>                 <NA>
  ESTIMATED.QUANTITY.10~2 TYPE.MEASUREMENT DRUG.INVOLVEMENT.2 DRUG.INVOLVEMENT.3
  <chr>                   <chr>            <chr>              <chr>
1 <NA>                    <NA>             <NA>               <NA>
  WINDOW.UCR.OFFENSE.CODE.1 WINDOW.UCR.OFFENSE.CODE.2 WINDOW.UCR.OFFENSE.CODE.3
  <chr>                     <chr>                     <chr>
1 <NA>                      <NA>                      <NA>
# i abbreviated names: 1: NUMBER.OF.STOLEN.MOTOR.VEHICLES, 2: ESTIMATED.QUANTITY.1000THS
# i 7 more variables: WINDOW.UCR.OFFENSE.CODE.4 <chr>, WINDOW.UCR.OFFENSE.CODE.5 <chr>, WIND
```

This incident involved something stolen (`TYPE.PROPERTY.LOSS.ETC` equal to 7). The thing that was stolen was "Portable Electronic Communications" (`PROPERTY.DESCRIPTION` code is 75), which most likely means a mobile phone. The estimated value of the phone was $150.

Importantly, the value of property stored in `VALUE.OF.PROPERTY` is not always the actual value. NIBRS codes certain special values in `VALUE.OF.PROPERTY`.

- 1 = the value of the property is unknown
- NA = the property was seized (`TYPE.PROPERTY.LOSS.ETC` is 6) and the seized property was drugs/narcotics (`PROPERTY.DESCRIPTION` is 10)
- the value of drugs will be present if they were damaged, destroyed, or stolen when the offense is not a narcotics violation, such as arson, burglary, or robbery.

3.4 Victim segment

We repeat the same process for the victim segment.

```r
i <- grep('LEVEL "04"', fmtXL$DataField)
j <- grep('LEVEL "05"', fmtXL$DataField)

fmt <- fmtXL |>
  slice((i+1):(j-3)) |>
  filter(!is.na(Position) &
           !(Position %in% c("37-66","74-77","79-83","84-123")))

fmtType <- fmt$TypeLength |>
   substring(1,1) |>
   case_match("A"~"c","N"~"n")
fmtWidth <- fmt$TypeLength |>
   substring(2) |>
   as.numeric()
fmtNames <- fmt$Description |>
   strsplit(split = " - ", fixed = TRUE) |>
   sapply(head, n=1) |>
   make.names() |>
   gsub("\\.+(\\.|$)", "\\1", x=_)

nibrsVic04 <- read_fwf("2023-04.txt.gz",
                       col_positions = fwf_widths(fmtWidth),
                       col_type=paste(fmtType,collapse="")) |>
   rename_with(~fmtNames) |>
   rename(ORI     = ORIGINATING.AGENCY.IDENTIFIER.ORI,
          SEGMENT = SEGMENT.LEVEL,
          STATE   = NUMERIC.STATE.CODE)
```

Let's pull up the victim record for the Fairbanks, Alaska incident 2300003 that we previously examined.

```r
nibrsVic04 |>
   filter(ORI=="AK0010200" & INCIDENT.NUMBER=="23000003") |>
   print(width=360)
```

```
# A tibble: 1 x 44
  SEGMENT STATE ORI       INCIDENT.NUMBER INCIDENT.DATE VICTIM.SEQUENCE.NUMBER
  <chr>   <chr> <chr>     <chr>           <chr>                          <dbl>
1 04      50    AK0010200 23000003        20230101                           1
  UCR.OFFENSE.CODE.1 UCR.OFFENSE.CODE.2 UCR.OFFENSE.CODE.3 UCR.OFFENSE.CODE.4
  <chr>              <chr>              <chr>              <chr>
```

```
1 23D                 <NA>                <NA>                <NA>
  UCR.OFFENSE.CODE.5 UCR.OFFENSE.CODE.6 UCR.OFFENSE.CODE.7 UCR.OFFENSE.CODE.8
  <chr>               <chr>               <chr>               <chr>
1 <NA>                <NA>                <NA>                <NA>
  UCR.OFFENSE.CODE.9 UCR.OFFENSE.CODE.10 TYPE.OF.VICTIM AGE.OF.VICTIM
  <chr>               <chr>               <chr>          <chr>
1 <NA>                <NA>                I              61
  SEX.OF.VICTIM RACE.OF.VICTIM ETHNICITY.OF.VICTIM RESIDENT.STATUS.OF.VICTIM
  <chr>          <chr>          <chr>               <chr>
1 M              W              N                   R
# i 22 more variables: CIRCUMSTANCE.1 <chr>, CIRCUMSTANCE.2 <chr>, ADDITIONAL.JUSTIFIABLE.HOM
#   RELATIONSHIP.5 <chr>, RELATIONSHIP.6 <chr>, RELATIONSHIP.7 <chr>, RELATIONSHIP.8 <chr>, R
```

Here we see that `TYPE.OF.VICTIM` is I, meaning that the victim was an individual (instead of a business, government, society, or some other institutional victim). The individual is a 61 year old non-Hispanic white male. No other information is available for this victim, but police can record the relationship between the victim and offender, whether the victim sustained any injuries, what drew the officer to the victim (such as a traffic stop or responding to a call for service), and whether the victim killed the offender in self-defense.

3.5 Offender segment

A few more segments to go. Now we will read in the data on the offenders.

```r
i <- grep('LEVEL "05"', fmtXL$DataField)
j <- grep('LEVEL "06"', fmtXL$DataField)

fmt <- fmtXL |>
  slice((i+1):(j-3)) |>
  filter(!is.na(Position))

fmtType <- fmt$TypeLength |>
   substring(1,1) |>
   case_match("A"~"c","N"~"n")
fmtWidth <- fmt$TypeLength |>
   substring(2) |>
   as.numeric()
fmtNames <- fmt$Description |>
   strsplit(split = " - ", fixed = TRUE) |>
   sapply(head, n=1) |>
   make.names() |>
   gsub("\\.+(\\.|$)", "\\1", x=_)

nibrsOff05 <- read_fwf("2023-05.txt.gz",
                       col_positions = fwf_widths(fmtWidth),
                       col_type=paste(fmtType,collapse="")) |>
   rename_with(~fmtNames) |>
   rename(ORI     = ORIGINATING.AGENCY.IDENTIFIER.ORI,
          SEGMENT = SEGMENT.LEVEL,
          STATE   = NUMERIC.STATE.CODE)
```

In some years the offender data has some text in the `OFFENDER.SEQUENCE.NUMBER` for some reason, like setting the `OFFENDER.SEQUENCE.NUMBER` to "##". We told `read_fwf()` to expect a numeric value for this column, marked by the "n" in the `fmtType` column here.

```r
data.frame(fmtNames, fmtType)
```

```
                            fmtNames fmtType
1                      SEGMENT.LEVEL       c
2                 NUMERIC.STATE.CODE       c
3  ORIGINATING.AGENCY.IDENTIFIER.ORI       c
4                    INCIDENT.NUMBER       c
5                      INCIDENT.DATE       c
6           OFFENDER.SEQUENCE.NUMBER       n
7                    AGE.OF.OFFENDER       c
8                    SEX.OF.OFFENDER       c
9                   RACE.OF.OFFENDER       c
10             Ethnicity.OF.OFFENDER       c
```

If this should happen, you can explore the problem a little to find the problem in the original data using `problems(nibrsOff05)`. My preferred solution is just to read all columns in as text and worry about fixing the non-numeric values later. Here I set all of the value of `col_type` to character.

```r
nibrsOff05 <- read_fwf("2023-05.txt.gz",
                       col_positions = fwf_widths(fmtWidth),
                       col_type="cccccccccc") |>
   rename_with(~fmtNames) |>
   rename(ORI=ORIGINATING.AGENCY.IDENTIFIER.ORI,
          SEGMENT=SEGMENT.LEVEL,
          STATE=NUMERIC.STATE.CODE)
```

We have no description of the offender in this case.

```
nibrsOff05 |>
   filter(ORI=="AK0010200" & INCIDENT.NUMBER=="23000003") |>
   print(width=360)
```

```
# A tibble: 1 x 10
  SEGMENT STATE ORI       INCIDENT.NUMBER INCIDENT.DATE OFFENDER.SEQUENCE.NUMBER
  <chr>   <chr> <chr>     <chr>           <chr>                            <dbl>
1 05      50    AK0010200 23000003        20230101                             0
  AGE.OF.OFFENDER SEX.OF.OFFENDER RACE.OF.OFFENDER Ethnicity.OF.OFFENDER
  <chr>           <chr>           <chr>            <chr>
1 <NA>            <NA>            <NA>             <NA>
```

3.6 Arrestee segment

If the reporting agency arrests an offender for the reported offense after submitting the initial report, the agency later submits an updated arrestee segment as an update to the initial report. The NIBRS User's Manual notes that as cases develop, police can update, delete, or add to previously submitted records. However, the submission date cuts off on March 15 of subsequent year.

```
# select only the rows for the arrestee segment
i <- grep('LEVEL "06"', fmtXL$DataField)
j <- grep('LEVEL "07"', fmtXL$DataField)

fmt <- fmtXL |>
  slice((i+2):(j-3)) |>
  filter(!is.na(Position) &
         !(Position %in% c("61-66","75-104")))

fmtType <- fmt$TypeLength |>
   substring(1,1) |>
   case_match("A"~"c","N"~"n")
fmtWidth <- fmt$TypeLength |>
   substring(2) |>
   as.numeric()
fmtNames <- fmt$Description |>
   strsplit(split = " - ", fixed = TRUE) |>
   sapply(head, n=1) |>
   make.names() |>
   gsub("\\.+(\\.|$)", "\\1", x=_)

nibrsArr06 <- read_fwf("2023-06.txt.gz",
                       col_positions = fwf_widths(fmtWidth),
                       col_type=paste(fmtType,collapse="")) |>
   rename_with(~fmtNames) |>
   rename(ORI     = ORIGINATING.AGENCY.IDENTIFIER.ORI,
          SEGMENT = SEGMENT.LEVEL,
          STATE   = NUMERIC.STATE.CODE)
```

No arrests were made in this case. You probably already suspected this since there was no description of an offender in this case either.

```
nibrsArr06 |>
   filter(ORI=="AK0010200" & INCIDENT.NUMBER=="23000003") |>
   count()
```

```
# A tibble: 1 x 1
      n
  <int>
1     0
```

Group B arrestees are kept in a separate data file. Group B offenses are almost all crimes against society, such as bad checks, non-violent family offenses, curfew/loitering/vagrancy, liquor law violations, disorderly conduct, peeping Tom, DUI, trespassing, drunkenness. These offenses are only reported if the police made an arrest. That is, someone calls the police reporting some disorderly conduct, the police do not record that in NIBRS unless they end up making an arrest as a result of the call.

```
i <- grep('LEVEL "07"', fmtXL$DataField)

fmt <- fmtXL |>
  slice(-(1:i)) |>
  filter(!is.na(Position) &
         !(Position %in% "44-49"))

fmtType <- fmt$TypeLength |>
   substring(1,1) |>
   case_match("A"~"c","N"~"n")
fmtWidth <- fmt$TypeLength |>
   substring(2) |>
   as.numeric()
fmtNames <- fmt$Description |>
   strsplit(split = " - ", fixed = TRUE) |>
   sapply(head, n=1) |>
   make.names() |>
   gsub("\\.+(\\.|$)", "\\1", x=_)

nibrsArB07 <- read_fwf("2023-07.txt.gz",
                       col_positions = fwf_widths(fmtWidth),
                       col_type=paste(fmtType,collapse="")) |>
   rename_with(~fmtNames) |>
   rename(ORI     = ORIGINATING.AGENCY.IDENTIFIER.ORI,
          SEGMENT = SEGMENT.LEVEL,
          STATE   = NUMERIC.STATE.CODE)
```

The Fairbanks case we traced through the other NIBRS tables was not a Group B offense, so it will have no records in this table. Instead, let's examine the first two Fairbanks Group B offenses reported. I have gone ahead and joined the Group B arrestee table with the `OffenseLookup` so the offenses are easier to interpret.

```
nibrsArB07 |>
   filter(ORI=="AK0010200") |>
   head(2) |>
   left_join(OffenseLookup, by=join_by(UCR.GROUP.B.ARREST.OFFENSE.CODE==UCRCode)) |>
   print(width=360)
```

```
# A tibble: 2 x 20
  SEGMENT STATE ORI       ARREST.TRANSACTION.INCIDENT.NUMBER ARREST.DATE
  <chr>   <chr> <chr>     <chr>                              <chr>
1 07      50    AK0010200 50901                              20230206
2 07      50    AK0010200 50908                              20230102
  ARRESTEE.SEQUENCE.NUMBER CITY.SUBMISSION TYPE.OF.ARREST
                     <dbl> <chr>           <chr>
1                        1 <NA>            T
2                        1 <NA>            O
  UCR.GROUP.B.ARREST.OFFENSE.CODE WEAPON.1 AUTOMATIC.INDICATOR.1 WEAPON.2
  <chr>                           <chr>    <chr>                 <chr>
1 90F                             01       <NA>                  <NA>
2 90C                             01       <NA>                  <NA>
  AGE.OF.ARRESTEE SEX.OF.ARRESTEE RACE.OF.ARRESTEE ETHNICITY.OF.ARRESTEE
  <chr>           <chr>           <chr>            <chr>
```

```
1 45               M                W                N
2 35               F                I                N
  RESIDENT.STATUS.OF.ARRESTEE DISPOSITION.OF.ARRESTEE.UNDER.18 CrimeCat     Crime
  <chr>                       <chr>                            <chr>        <chr>
1 R                           <NA>                             Family Off~  Fami~
2 N                           <NA>                             Disorderly~  Diso~
```

Police arrested the first individual listed here based on a warrant for a prior incident (`TYPE.OF.ARREST` is "T") for non-violent family offenses (e.g. neglect, not paying alimony). `WEAPON.1` equal to "01" indicates that the individual was unarmed. Age, race, sex information is also provided in this arrestee table.

3.7 Batch Header segment

The final segment we will read in is the batch header file. This file contains basic information about the agencies reporting to NIBRS. It contains one record for each law enforcement agency.

```r
fmt <- read_excel("NIBRS Records Description updated.xlsx",
                  skip=4) |>
       data.frame() |>
       filter(!is.na(Position) &
              !(Position %in% c("106-225", # data on county can repeat 5x
                                "234-269", # indicators for 12 months
                                "234","235","236", # detailed January reporting
                                "270-284"))) # county codes up to 5x

fmtType <- fmt$Type..Length |>
   substring(1,1) |>
   case_match("A"~"c","N"~"n")
fmtWidth <- fmt$Type..Length |>
   substring(2) |>
   as.numeric()
fmtNames <- fmt$Description |>
   strsplit(split = " - ", fixed = TRUE) |>
   sapply(head, n=1) |>
   make.names() |>
   gsub("\\.+(\\.|$)", "\\1", x=_)

nibrsBH <- read_fwf("2023-BH.txt.gz",
                    col_positions = fwf_widths(fmtWidth),
                 col_type=paste(fmtType,collapse="")) |>
   rename_with(~fmtNames) |>
   rename(ORI      = ORIGINATING.AGENCY.IDENTIFIER.ORI,
          SEGMENT = SEGMENT.LEVEL,
          STATE   = NUMERIC.STATE.CODE)
```

Some particularly important columns are `DATE.ORI.WENT.NIBRS` containing the date when the agency first reported data to NIBRS, the `AGENCY.INDICATOR` describing what type of law enforcement agency it is (e.g. city, county, state, college, tribal) or whether the agency relies on another to file its NIBRS reports (such as a transit police that reports through the municipal police). The batch header contains an estimate of the size of the population that the agency covers, but breaks down the reporting by county. Oklahoma City, for example, intersects with four different counties, so that the Oklahoma City Police Department reports their population separately for each.

```
nibrsBH |>
   filter(ORI=="OK0550600") |>
   select(ORI, CITY.NAME, STATE.ABBREVIATION,
          starts_with("CURRENT.POPULATION")) |>
   print(width=360)
```

```
# A tibble: 1 x 8
  ORI       CITY.NAME     STATE.ABBREVIATION CURRENT.POPULATION.1
  <chr>     <chr>         <chr>                             <dbl>
1 OK0550600 OKLAHOMA CITY OK                               532970
  CURRENT.POPULATION.2 CURRENT.POPULATION.3 CURRENT.POPULATION.4
                 <dbl>                <dbl>                <dbl>
1                88049                79665                   80
  CURRENT.POPULATION.5
                 <dbl>
1                    0
```

I find it helpful to have a total population covered.

```
nibrsBH <- nibrsBH |>
   mutate(TOTAL.POP=rowSums(across(starts_with("CURRENT.POP")), na.rm=TRUE))
```

The batch header also contains information on whether the agency regularly reports their data. For example, here is the record for San Jose, California.

```r
nibrsBH |> filter(ORI=="CA0431300") |>
   select(ORI, CITY.NAME, STATE.ABBREVIATION, DATE.ORI.WENT.NIBRS,
          NUMBER.OF.MONTHS.REPORTED, JANUARY:DECEMBER) |>
   print(width=1000)
```

```
# A tibble: 1 x 17
  ORI       CITY.NAME STATE.ABBREVIATION DATE.ORI.WENT.NIBRS
  <chr>     <chr>     <chr>              <chr>
1 CA0431300 SAN JOSE  CA                 20230401
  NUMBER.OF.MONTHS.REPORTED JANUARY FEBRUARY MARCH APRIL MAY   JUNE  JULY
  <chr>                     <chr>   <chr>    <chr> <chr> <chr> <chr> <chr>
1 07                        NNN     NNN      NNN   NYN   NYN   NYN   NYN
  AUGUST SEPTEMBER OCTOBER NOVEMBER DECEMBER
  <chr>  <chr>     <chr>   <chr>    <chr>
1 NYN    NYN       NYN     NNN      NNN
```

This record shows that they started reporting to NIBRS in April 2023 and reported for a total of 7 months. The `JANUARY` to `DECEMBER` columns describe more precisely what the agency submitted. The "NNN" indicates no reporting of any kind to NIBRS, occurring January, February, and March and then again in November and December. The "NYN" pattern means that the agency did not report (the first "N") a "Zero Report" (a formal notice of no crime reports), did report Group A or Group B offenses (the "Y"), and did not report any window records (the final N, meaning there was no partial reporting, like an arrest without an associated offense report).

Let's explore which agencies are reporting to NIBRS so we have a better sense of what NIBRS includes and excludes.

```r
# NYPD, LAPD, and Nashville PD
nibrsBH |>
   filter(ORI %in% c("NY0303000","CA0194200","TN0190100")) |>
   select(ORI, CITY.NAME, STATE.ABBREVIATION, DATE.ORI.WENT.NIBRS,
          NUMBER.OF.MONTHS.REPORTED, JANUARY:DECEMBER) |>
   print(width=Inf)
```

```
# A tibble: 3 x 17
  ORI       CITY.NAME   STATE.ABBREVIATION DATE.ORI.WENT.NIBRS
  <chr>     <chr>       <chr>              <chr>
1 CA0194200 LOS ANGELES CA                 <NA>
2 NY0303000 NEW YORK    NY                 20230101
3 TN0190100 NASHVILLE   TN                 19991001
```

```
  NUMBER.OF.MONTHS.REPORTED JANUARY FEBRUARY MARCH APRIL MAY   JUNE  JULY
  <chr>                     <chr>   <chr>    <chr> <chr> <chr> <chr> <chr>
1 00                        NNN     NNN      NNN   NNN   NNN   NNN   NNN
2 12                        NYN     NYN      NYN   NYN   NYN   NYN   NYN
3 12                        NYN     NYN      NYN   NYN   NYN   NYN   NYN
  AUGUST SEPTEMBER OCTOBER NOVEMBER DECEMBER
  <chr>  <chr>     <chr>   <chr>    <chr>
1 NNN    NNN       NNN     NNN      NNN
2 NYN    NYN       NYN     NYN      NYN
3 NYN    NYN       NYN     NYN      NYN
```

Los Angeles PD has never reported to NIBRS, New York City started their NIBRS reporting in 2023, and Nashville has been reporting to NIBRS for 25 years, since 1999 and continues to report its crime data every month.

Which states seem to have good reporting?

```r
nibrsBH |>
  filter(is.na(COVERED.BY.ORI)) |> # remove those reporting through another
  count(STATE.ABBREVIATION, AGENCY.NIBRS.FLAG) |>
  pivot_wider(names_from = AGENCY.NIBRS.FLAG,
              values_from = n,
              values_fill = 0) |>
  rename(inNIBRS=A, notInNIBRS=`NA`) |>
  print(n=Inf)
```

```
# A tibble: 54 x 3
   STATE.ABBREVIATION inNIBRS notInNIBRS
   <chr>                <int>      <int>
 1 AK                      32          7
 2 AL                     415         31
 3 AR                     313          3
 4 AZ                     100         26
 5 CA                     634        225
 6 CO                     245          4
 7 CT                     108          0
 8 DC                       2          1
 9 DE                      63          0
10 FL                     133        625
11 GA                     499        164
12 GM                       1          1
13 HI                       4          0
```

```
14 IA                      286          3
15 ID                      112          0
16 IL                      635        288
17 IN                      260         97
18 KS                      406          2
19 KY                      436          1
20 LA                      165         70
21 MA                      398         22
22 MD                      139         18
23 ME                      131          0
24 MI                      634          2
25 MN                      409          2
26 MO                      537         62
27 MS                      168        108
28 MT                      115          0
29 NB                      283          4
30 NC                      442         93
31 ND                      111          0
32 NH                      216          5
33 NJ                      346        232
34 NM                      104         26
35 NV                       61         11
36 NY                      185        405
37 OH                      813         42
38 OK                      462          0
39 OR                      234          3
40 PA                      171       1306
41 PR                        0          1
42 RI                       48          1
43 SC                      513          1
44 SD                      141          0
45 TN                      411          1
46 TX                     1393         84
47 UT                      144          2
48 VA                      420          0
49 VI                        0          2
50 VT                       87          0
51 WA                      270          0
52 WI                      394         64
53 WV                      426          4
54 WY                       53         17
```

The states that have a large number of non-reporting agencies are California, Florida, Georgia, Illinois, New Jersey, New York, and Pennsylvania. These are rather large states. In the event that the non-reporting agencies are small, it is probably better to assess the percentage of the population covered by NIBRS rather than counting non-reporting agencies.

```
nibrsBH |>
  filter(is.na(COVERED.BY.ORI)) |> # remove those reporting through another
  group_by(STATE.ABBREVIATION, AGENCY.NIBRS.FLAG) |>
  summarize(populationCovered=sum(TOTAL.POP)) |>
  pivot_wider(names_from = AGENCY.NIBRS.FLAG,
              values_from = populationCovered,
              values_fill = 0) |>
  rename(inNIBRS=A, notInNIBRS=`NA`) |>
  mutate(percentCovered=round(100*inNIBRS/(inNIBRS+notInNIBRS))) |>
  arrange(percentCovered) |>
  head(10)
```

```
# A tibble: 10 x 4
# Groups:   STATE.ABBREVIATION [10]
   STATE.ABBREVIATION  inNIBRS notInNIBRS percentCovered
   <chr>                 <dbl>      <dbl>          <dbl>
 1 PR                        0    3205691              0
 2 VI                        0      83265              0
 3 FL                  9583627   13027099             42
 4 PA                  5605372    7356311             43
 5 AK                   445330     288076             61
 6 AZ                  4664949    2766395             63
 7 CA                 25034801   13930392             64
 8 MS                  1905988    1033702             65
 9 NJ                  6351608    2939233             68
10 NY                 13614572    5956644             70
```

The states with large populations that NIBRS does not cover are Florida, Pennsylvania, and California.

3.8 Saving our work

Save the NIBRS 2023 data in its R format. If you ever want to reload the data, there will be no need to rerun all the previous steps. You can just use `load("nibrs2023.RData")` to retrieve the data. It will be quite large and can take a few minutes to reload.

```
save(nibrsAdmin01, nibrsCrm02, nibrsProp03, nibrsVic04, nibrsOff05,
     nibrsArr06, nibrsArB07, nibrsBH,
     fmtXL,
     file="nibrs2023.RData",
     compress = TRUE)
```

4 Examples using NIBRS

For several of these examples we are going to need to work with dates. So, let's start by converting all the dates to proper date formats.

```
nibrsCrm02 <- nibrsCrm02 |>
   mutate(INCIDENT.DATE = ymd(INCIDENT.DATE))
nibrsArr06 <- nibrsArr06 |>
   mutate(INCIDENT.DATE = ymd(INCIDENT.DATE),
          ARREST.DATE = ymd(ARREST.DATE))
```

4.1 What percentage of burglaries result in an arrest?

Remember that arrest data only go through March 15 of the subsequent year. Note that the arrest will show up only if the agency records the arrest by March 15, 2024. Let's start this example by finding the right UCR codes for burglary. Note that `str_detect()` is equivalent to `grepl()`. You may use either of them in such `filter()` statements.

```
OffenseLookup |> filter(str_detect(Crime, "[Bb]urglary"))
```

```
  UCRCode                     CrimeCat                         Crime
1     220 Burglary/Breaking & Entering Burglary/Breaking & Entering
```

We will use UCR code 220 to select all the burglary offenses and check that `OFFENSE.ATTEMPTED.COMPLETED` equals "C" to eliminate the attempted burglaries.

```
nibrsCrm02 |>
  filter(UCR.OFFENSE.CODE=="220" &             # burglary offense code
         OFFENSE.ATTEMPTED.COMPLETED=="C") |> # completed
  select(ORI, INCIDENT.NUMBER) |> # incidents are only unique within ORI
  distinct() |> # remove any duplicate ORI/INCIDENT.NUMBER combos
  # compress arrest data to ORI, INCIDENT.NUMBER, and first arrest
  left_join(nibrsArr06 |>
               group_by(ORI, INCIDENT.NUMBER) |>
               summarize(dateFirstArrest = min(ARREST.DATE, na.rm=TRUE)) |>
               ungroup(),
            by = join_by(ORI, INCIDENT.NUMBER)) |>
  summarize(pctArrest = mean(!is.na(dateFirstArrest)))
```

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

```
# A tibble: 1 x 1
  pctArrest
      <dbl>
1     0.136
```

4.2 What percentage of crimes are cleared with an arrest by month in which the offense occurred?

```
nibrsCrm02 |>
  mutate(INCIDENT.MONTH=month(INCIDENT.DATE, label=TRUE)) |>
  select(ORI, INCIDENT.NUMBER, INCIDENT.MONTH) |>
  distinct() |>
  left_join(nibrsArr06 |>
                group_by(ORI, INCIDENT.NUMBER) |>
                summarize(dateFirstArrest = min(ARREST.DATE, na.rm=TRUE)) |>
                ungroup(),
             by = join_by(ORI, INCIDENT.NUMBER)) |>
  group_by(INCIDENT.MONTH) |>
  summarize(pctArrest = mean(!is.na(dateFirstArrest)))
```

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

```
# A tibble: 12 x 2
   INCIDENT.MONTH pctArrest
   <ord>              <dbl>
 1 Jan                0.246
 2 Feb                0.248
 3 Mar                0.249
 4 Apr                     0.248
 5 May                     0.244
 6 Jun                     0.241
 7 Jul                     0.241
 8 Aug                     0.242
 9 Sep                     0.244
10 Oct                     0.238
11 Nov                     0.243
12 Dec                     0.242
```

4.3 What is the range of times to arrest for crimes resulting in arrest?

This next example shows the issue with the March 15 cutoff for reporting to NIBRS. Have a look at the average time and maximum time to arrest by month of the incident. The results seem to suggest that crimes resulting in arrest from January take a lot longer to produce the arrest than crimes in December. However, those January crimes have 15 months to produce an arrest (January 2023 to March 15, 2024) while the December arrests have only three months to produce an arrest before NIBRS reporting closes for the year. The observed pattern is entirely an artifact of the NIBRS data collection indicating that calculating clearance rates or time to clearance requires more care.

```r
nibrsArr06 |>
   group_by(ORI, INCIDENT.NUMBER) |>
   summarize(dateFirstArrest = min(ARREST.DATE, na.rm=TRUE)) |>
   ungroup() |>
   left_join(nibrsCrm02 |>
                select(ORI, INCIDENT.NUMBER, INCIDENT.DATE) |>
                distinct(),
             by = join_by(ORI, INCIDENT.NUMBER)) |>
   mutate(timeToArrest = difftime(dateFirstArrest, INCIDENT.DATE,
                                  units="days"),
          INCIDENT.MONTH = month(INCIDENT.DATE, label=TRUE)) |>
   group_by(INCIDENT.MONTH) |>
   summarize(mean = mean(timeToArrest, na.rm=TRUE),
             max  = max(timeToArrest,  na.rm=TRUE)) |>
   print(width=Inf)
```

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

```
# A tibble: 12 x 3
```

```
   INCIDENT.MONTH mean           max
   <ord>          <drtn>         <drtn>
 1 Jan            11.864666 days 451 days
 2 Feb             9.946931 days 423 days
 3 Mar             9.304757 days 398 days
 4 Apr             8.638813 days 361 days
 5 May             8.325869 days 332 days
 6 Jun             7.769507 days 305 days
 7 Jul             7.028547 days 275 days
 8 Aug             6.381894 days 249 days
 9 Sep             5.575103 days 213 days
10 Oct             4.851006 days 185 days
11 Nov             3.846566 days 155 days
12 Dec             2.895208 days 125 days
```

For a more fair comparison across months we can look at the percentage of crimes that result in an arrest within two months of the crime incident date. Technically, crimes on December 31st have until March 15th to have any arrests recorded (75 days), but I am going to set the threshold to within 60 days just in case it takes a few days for arrests in February for December crimes to get posted to NIBRS. Here we see that the percentage of crimes cleared with an arrest within 60 days is nearly constant across the months of the year.

```r
nibrsCrm02 |>
   select(ORI, INCIDENT.NUMBER, INCIDENT.DATE) |>
   distinct() |>
   left_join(nibrsArr06 |>
                group_by(ORI, INCIDENT.NUMBER, ARREST.DATE) |>
                summarize(dateFirstArrest = min(ARREST.DATE, na.rm=TRUE)) |>
                ungroup(),
             by = join_by(ORI, INCIDENT.NUMBER)) |>
   mutate(timeToArrest = difftime(dateFirstArrest, INCIDENT.DATE,
                                  units="days"),
          arrest2Months = as.numeric(!is.na(timeToArrest) & timeToArrest <= 60),
          INCIDENT.MONTH = month(INCIDENT.DATE, label=TRUE)) |>
   group_by(INCIDENT.MONTH) |>
   summarize(mean=mean(arrest2Months, na.rm=TRUE)) |>
   ungroup() |>
   print(width=Inf)
```

```
`summarise()` has grouped output by 'ORI', 'INCIDENT.NUMBER'. You can override
using the `.groups` argument.
```

```
# A tibble: 12 x 2
   INCIDENT.MONTH  mean
   <ord>          <dbl>
 1 Jan            0.234
 2 Feb            0.238
 3 Mar            0.239
 4 Apr            0.239
 5 May            0.235
 6 Jun            0.232
 7 Jul            0.233
 8 Aug            0.235
 9 Sep            0.238
10 Oct            0.233
11 Nov            0.240
12 Dec            0.241
```

5 Exercises

3. How many offenses are shootings? That is, the offense was aggravated assault (13A), and the offense was completed (C), and the weapon involved was a firearm (11, 12, 13, 14, 15).
4. What is the race distribution of shooting victims?
5. How many cases are recorded as justifiable homicides? Note the special codes in the NIBRS documentation file.

Code	Description
20	Criminal killed by private citizen
21	Criminal killed by police officer
A	Criminal attacked police officer and that officer killed criminal
B	Criminal attacked fellow police officer and criminal killed by another police officer
C	Criminal attacked a civilian
D	Criminal attempted flight from a crime
E	Criminal killed in commission of a crime
F	Criminal resisted arrest
G	Unable to determine/not enough information

Other data sources show that the total number of fatal police shooting victims is about 1,000 per year.

6. Does NIBRS indicate that retail theft is rising? This exercise will involve a lot more work, including acquiring NIBRS data from multiple years. First, listen to the story "Is retail theft really rising?" featuring Kai Ryssdal and Livi Burdette recorded on September 11, 2023. In the discussion, one reporter seems to indicate that there are no national data on retail theft. Now that you have worked through this script and worked with NIBRS, you can see that retail theft is a UCR code reportable to NIBRS. Now some stores might not report retail theft to the police, but such non-reporting plagues all crime types. But you can evaluate whether reports of retail theft are increasing or decreasing over time. When examining trends over the last several years, make sure to consider that some agencies, like NYPD, just started reporting in 2023. Best to exclude them if you wish to compare trends from 2021 to the present. So, do the NIBRS data indicate that retail theft is rising?

6 Answers to the exercises

1. How many Group A offenses were reported to the Philadelphia Police Department (ORI PAPEP0000)?

```r
nibrsCrm02 |> filter(ORI=="PAPEP0000") |> count()
```

```
# A tibble: 1 x 1
       n
   <int>
1 170168
```

2. What is the most commonly reported crime in Philadelphia? Use the OffenseLookup to replace the UCR codes with crime descriptions.

```r
nibrsCrm02 |>
   filter(ORI=="PAPEP0000") |>
   count(UCR.OFFENSE.CODE) |>
   slice_max(n) |>
   left_join(OffenseLookup,
             by=join_by(UCR.OFFENSE.CODE==UCRCode)) |>
   select(n, Crime)
```

```
# A tibble: 1 x 2
      n Crime
  <int> <chr>
1 31068 Destruction/Damage/Vandalism of Property
```

3. How many offenses are shootings?

```
nibrsCrm02 |>
  filter(UCR.OFFENSE.CODE=="13A" &
           OFFENSE.ATTEMPTED.COMPLETED=="C" &
           (WEAPON.FORCE.1 %in% 11:15 |
            WEAPON.FORCE.2 %in% 11:15 |
            WEAPON.FORCE.3 %in% 11:15)) |>
  count()
```

```
# A tibble: 1 x 1
       n
   <int>
1 181371
```

4. What is the race distribution of shooting victims?

```
nibrsCrm02 |>
  filter(UCR.OFFENSE.CODE=="13A" &
         OFFENSE.ATTEMPTED.COMPLETED=="C" &
         (WEAPON.FORCE.1 %in% 11:15 |
          WEAPON.FORCE.2 %in% 11:15 |
          WEAPON.FORCE.3 %in% 11:15)) |>
  distinct() |>
  left_join(nibrsVic04,
            by=join_by(ORI, INCIDENT.NUMBER)) |>
  group_by(RACE.OF.VICTIM, ETHNICITY.OF.VICTIM) |>
  count()  |>
  pivot_wider(names_from = ETHNICITY.OF.VICTIM,
              values_from = n,
              values_fill = 0) |>
  print(n=Inf)
```

```
# A tibble: 7 x 5
# Groups:   RACE.OF.VICTIM [7]
  RACE.OF.VICTIM     H      N     U  `NA`
  <chr>          <int>  <int> <int> <int>
```

```
1 A                  165   2657   306   261
2 B                 1227 112878 11398 19727
3 I                  128   1726   271   434
4 P                  140    351    63    28
5 U                 2530   1456  5639  2046
6 W                46833  57513  7952 13205
7 <NA>                 0      0     0 30829
```

Here's a little more complete version that makes sure the victim in the incident is one of the shooting victims and tabulates Hispanic as a race category. Some offenses might involve an offender hitting one person and shooting another. This code makes sure we just count the race of those actually shot.

```
nibrsCrm02 |>
  filter(UCR.OFFENSE.CODE=="13A" &
         OFFENSE.ATTEMPTED.COMPLETED=="C" &
           (WEAPON.FORCE.1 %in% 11:15 |
            WEAPON.FORCE.2 %in% 11:15 |
            WEAPON.FORCE.3 %in% 11:15)) |>
  select(ORI, INCIDENT.NUMBER) |>
  distinct() |>
  left_join(nibrsVic04 |>
              filter(UCR.OFFENSE.CODE.1=="13A" | UCR.OFFENSE.CODE.2 =="13A" |
                     UCR.OFFENSE.CODE.3=="13A" | UCR.OFFENSE.CODE.4 =="13A" |
                     UCR.OFFENSE.CODE.5=="13A" | UCR.OFFENSE.CODE.6 =="13A" |
                     UCR.OFFENSE.CODE.7=="13A" | UCR.OFFENSE.CODE.8 =="13A" |
                     UCR.OFFENSE.CODE.9=="13A" | UCR.OFFENSE.CODE.10=="13A") |>
              select(ORI,
                     INCIDENT.NUMBER,
                     RACE.OF.VICTIM,
                     ETHNICITY.OF.VICTIM),
            by=join_by(ORI, INCIDENT.NUMBER)) |>
  mutate(race2=case_when(ETHNICITY.OF.VICTIM=="H" ~ "H",
                         .default=RACE.OF.VICTIM)) |>
  group_by(race2) |>
  count()
```

```
# A tibble: 7 x 2
# Groups:   race2 [7]
  race2      n
  <chr>  <int>
1 A       2904
```

```
2 B     134553
3 H      47925
4 I       2271
5 P        412
6 U       7833
7 W      73097
```

5. How many cases are recorded as justifiable homicides?

Here's the first idea. We can count "criminal" incidents involving UCR code 09C, justifiable homicide.

```
nibrsCrm02 |>
  filter(UCR.OFFENSE.CODE=="09C") |>
  count()
```

```
# A tibble: 1 x 1
      n
  <int>
1   671
```

A second approach is to count victims, those killed by citizens (circumstance 20) and those killed by police (circumstance 21).

```
nibrsVic04 |> count(CIRCUMSTANCE.1)
```

```
# A tibble: 18 x 2
   CIRCUMSTANCE.1        n
   <chr>             <int>
 1 01               294313
 2 02                22595
 3 03                 2136
 4 04                 3066
 5 05                 1905
 6 06                99067
 7 07                   16
 8 08                 9620
 9 09               143124
10 10               184583
11 20                  443
12 21                  248
13 30                   38
```

```
14 31                     10
15 32                      2
16 33                    298
17 34                   1670
18 <NA>             13153518
```

```r
nibrsVic04 |> count(CIRCUMSTANCE.2)
```

```
# A tibble: 8 x 2
  CIRCUMSTANCE.2         n
  <chr>              <int>
1 02                   124
2 03                   135
3 04                   107
4 05                   121
5 06                  8209
6 08                  1001
7 09                  4629
8 <NA>            13902326
```

```r
sum(nibrsVic04$CIRCUMSTANCE.1 %in% 20:21)
```

```
[1] 691
```

```r
nibrsVic04 |> count(ADDITIONAL.JUSTIFIABLE.HOMICIDE.CIRCUMSTANCES)
```

```
# A tibble: 8 x 2
  ADDITIONAL.JUSTIFIABLE.HOMICIDE.CIRCUMSTANCES        n
  <chr>                                            <int>
1 A                                                  133
2 B                                                   37
3 C                                                  302
4 D                                                    4
5 E                                                  178
6 F                                                   13
7 G                                                   24
8 <NA>                                          13915961
```

Code	Description

20	Criminal killed by private citizen
21	Criminal killed by police officer
A	Criminal attacked police officer and that officer killed criminal
B	Criminal attacked fellow police officer and criminal killed by another police officer
C	Criminal attacked a civilian
D	Criminal attempted flight from a crime
E	Criminal killed in commission of a crime
F	Criminal resisted arrest
G	Unable to determine/not enough information

The following code offers a more complete description of who committed the justifiable homicide and the reason for the homicide.

```
nibrsVic04 |>
  filter(CIRCUMSTANCE.1 %in% c("20","21")) |>
  count(CIRCUMSTANCE.1, ADDITIONAL.JUSTIFIABLE.HOMICIDE.CIRCUMSTANCES) |>
  pivot_wider(names_from = ADDITIONAL.JUSTIFIABLE.HOMICIDE.CIRCUMSTANCES,
              values_from = n,
              values_fill = 0)
```

```
# A tibble: 2 x 8
  CIRCUMSTANCE.1     B     C     D     E     G     A     F
  <chr>          <int> <int> <int> <int> <int> <int> <int>
1 20                 1   276     1   150    15     0     0
2 21                36    26     3    28     9   133    13
```

6. Does NIBRS indicate that retail theft is rising? To examine trends we would need to load several years of NIBRS data. For now, let's just calculate the number of retail theft (23C) incidents reported to agencies that reported 12 months of NIBRS data.

```
nibrsCrm02 |>
   select(ORI, INCIDENT.NUMBER, UCR.OFFENSE.CODE) |>
   left_join(nibrsBH |> select(ORI,JANUARY:DECEMBER),
             by=join_by(ORI)) |>
   filter(UCR.OFFENSE.CODE=="23C" & # shoplifting
             if_all(JANUARY:DECEMBER, # all months had some report
                    function(x) x != "NNN")) |>
   count()
```

```
# A tibble: 1 x 1
        n
    <int>
1 945966
```