Criminal History Records Paper Step 1 (Data Management)

Steven J. Pierce

Contents

---

1 Purpose

This file imports copies of publicly archived data files (Campbell, 2019) into R, then performs data management to prepare a new data file for a descriptive paper about the criminal histories of suspected serial sexual perpetrators [1]. The data come from Detroit, MI.

Part of the analysis will require splitting each offender's criminal history into three periods:

- Before the offender's earliest known sexual assault kit (SAK).
- During a *testing window* that starts on the date of the offender's earliest known SAK. This window operationalizes the concept of timely forensic testing.
- After the end of the testing window.

[1] In this document, we use the terms perpetrator and offender interchangeably. Both terms should be interpreted as referring to individuals suspected of committing a crime but we occasionally omit that qualifier for the sake of brevity.

2 Defining Timely Testing

One purpose of forensic testing for SAKs is to prevent future crimes by facilitating identification and arrest of suspected offenders. However, forensic testing takes time to yield results. There are several steps that occur between SAK collection and law enforcement personnel being able to act on test results.

There are examples of testing taking as little as a week for cases deemed to be emergencies, but for non-emergency cases the testing time in Detroit had historically been much longer (typically 6-9 months, but 12-24 months was not uncommon) according to local stakeholders.

Michigan's 2014 Sexual Assault Kit Evidence Submission Act (PA227) provides a working definition of timely testing by stipulating the following requirements.

- Law enforcement agencies must take possession of an SAK within 14 days of receiving notice from a health care facility that it has been collected.
- Law enforcement agencies must submit the SAK to a lab for testing within 14 days of taking possession of it.
- Laboratories must complete testing within 90 days after receiving an SAK.

Adhering to those requirements would mean that law enforcement agencies should be able to start acting on forensic results no later than 118 days after SAK collection. Although some practitioners have reservations about the feasibility of meeting the timing requirements in this law, it is nevertheless the best working definition of timely testing available because it represents the outcome of a public policy process in which multiple stakeholders had a voice.

In this study, we adopted an operational definition for the testing window based on PA227's timing requirements. We defined the testing window as the first 118 days following the earliest SAK date associated with a suspected offender. It is unrealistic to expect that crimes committed during this period could be prevented by timely SAK testing because law enforcement personnel cannot act on information they don't have yet. Adopting a testing window definition based on the maximum duration consistent with the notion of timely testing is useful for initial descriptive analyses that aim to characterize what crimes might have been prevented if timely testing had been done on the SAKs in this sample.

3 Glossary

Here we define a few terms mentioned in this file.

- **Data frame.** This is a just a tabular data set comprised of rows (observations or cases) and columns (variables).
- **Tibble.** The tidyverse group of R packages create and use a modified type of data frame called tibbles. A tibble is still a tabular data set comprised of rows (observations or cases) and columns (variables).

4 Setup

Set global R chunk options (local chunk options will over-ride global options). The method for creating a size option that controls font size in code chunks and their text output is based on an answer to a question posted on stackoverflow.com.

```
# Create a custom chunk hook/option for controlling font size in chunk & output.
def.chunk.hook  <- knitr::knit_hooks$get("chunk")
knitr::knit_hooks$set(chunk = function(x, options) {
  x <- def.chunk.hook(x, options)
  ifelse(options$cfsize != "normalsize", paste0("\n \\", options$cfsize,"\n\n",
                                          x, "\n\n \\normalsize"), x)
  })

# Global chunk options (over-ridden by local chunk options)
knitr::opts_chunk$set(include  = TRUE, echo = TRUE, error = TRUE,
                      message = TRUE, warning = TRUE, cfsize = "footnotesize")

# Declare location of this script relative to the project root directory.
here::i_am(path = "inst/Step_01_Data_Mgt.Rmd")
```

```
## here() starts at S:/14-286/Analyses/SSACHR
```

Load R packages that we need to get additional functions.

```
library(here)             # for here()
library(plyr)             # For mapvalues()
```

```
##
## Attaching package: 'plyr'

## The following object is masked from 'package:here':
##
##     here
```

```
library(dplyr)            # for %>%, filter(), group_by(), & summarise()
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)            # for arrange(), filter(), group_by(), mutate(),
                          # spread(), summarise(), %>%, etc.
library(rmarkdown)        # for render()
library(knitr)            # for kable()
options(kableExtra.latex.load_packages = FALSE)
library(kableExtra)       # for kable_styling(), add_header_above(), column_spec(),
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
                          # collapse_rows(), and landscape()
library(descr)            # For freq().
options(descr.plot=FALSE) # Make freq() & crosstab() skip plots by default.
library(lubridate)        # For date conversion, eg. ymd(), time_length().
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(sjlabelled)       # For set_label(), get_label()
```

```
##
## Attaching package: 'sjlabelled'

## The following object is masked from 'package:dplyr':
##
##     as_label
```

```
library(haven)            # for read_spss()
```

```
##
## Attaching package: 'haven'
```

```
## The following objects are masked from 'package:sjlabelled':
##
##     as_factor, read_sas, read_spss, read_stata, write_sas, zap_labels
```

```
library(lattice)          # For xyplot(), bwplot(), etc.
library(psych)            # For describe()
library(car)              # For recode()
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(SSACHR)           # for git_report(), rvlabel(), which_latex()
library(vistime)          # for gg_vistime()
library(ggplot2)          # for theme()
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
## The following object is masked from 'package:sjlabelled':
##
##     as_label
```

```
library(RColorBrewer)     # for brewer.pal() and display.brewer.all()
```

5 Import Files

The SPSS files we will need to import, plus documentation about their contents and relationships between them, are available from an online archive (Campbell, 2019). After you have a local copy of this SSACHR git repository on your computer, place the downloaded data files in the subfolder *SSACHR/inst/extdata* without changing the file names to minimize the effort involved in using this script. Otherwise, you will need to update the file name references in the code below.

5.1 Core Criminal History Records (CHR) Files

Import the SPSS data files for the five core types of records comprising Michigan criminal history records (CHR): offenders (IDN), incidents (INC), arrest offenses (ARR), prosecutor charges (CHG), and judicial charges (JUD). See Campbell (2019) for details on these files.

The variables we are updating with *rvlabel()* all have user-missing values with labels that contain parentheses that are problematic when supplied to the *user.missing* argument in *freq()*, so we just remove the parentheses from the labels at import.

For the *ARR*, *CHG*, and *JUD* data, we use *dcode()* to convert the categorical variables *ACat12*, *CCat12*, and *JCat12* into a set of dummy-code variables with suffixes of the form *_x* where *x* includes the values 1-12 and 999. We also compute an new dummy code variable with the suffix *_Any* to code records where the relevant crime category was any of the 12 main codes treated as valid values instead of the user-missing code 999. These additional variables facilitate later record selection and aggregation operations.

We also add some additional dummy code variables that group crime categories 1-12 into four broader categories as shown below. These variables have the suffixes *_Sexual*, *_Violent*, *_Property*, and *_Other*. A particular arrest, charge, or adjudicated charge from categories 1-12 always belongs to both its own specific crime category and to one of those broader categories. This is a hierarchical categorization scheme.

- Sexual crimes, which include:
  - Criminal sexual conduct (CSC)
  - Sex crimes, non-CSC.
- Violent non-sexual crimes, which include:
  - Homicide
  - Assault - Domestic violence and/or stalking
  - Assault - Non-sexual, non-domestic violence
  - Robbery
  - Weapons
- Property crimes, which include:
  - Arson
  - Burglary
  - Larceny/Theft/Fraud
- Other crimes, which include:
  - Drug crimes
  - Traffic and ordinances

```
read_sav(here::here("inst/extdata/SSA_IDN_Offenders_2018-09-03.sav"),
         user_na = TRUE) %>%
  # Set variable label to match other core CHR files.
  var_labels(OID = "Offender ID (de-identified)") ->
  IDN

# Make attributes for OID match those in other core CHR files.
attr(IDN$OID, "format.spss") <- "A4"
```

```
read_sav(here::here("inst/extdata/SSA_INC_Incidents_Imputed_2018-09-03.sav"),
         user_na = TRUE) ->
  INC
```

```
read_sav(here::here("inst/extdata/SSA_ARR_Arrests_Imputed_2018-09-03.sav"),
         user_na = TRUE) %>%
  # Replace user-missing value labels.
  mutate(ACat12 = rvlabel(ACat12),
         ACat10 = rvlabel(ACat10),
         ACat04 = rvlabel(ACat04)) %>%
  # Dummy code crime category variables.
  dcode(x = ., y = factor(.$ACat12), stem = "ACat12") %>%
  # Dummy code any of the 12 main crime categories, set user-missing values to 0.
  mutate(ACat12_Any = if_else(ACat12 %in% 1:12, true = 1, false = 0)) %>%
  # Dummy code the combined sexual crimes categories, set user-missing values to 0.
  mutate(ACat12_Sexual = if_else(ACat12 %in% c(5, 10), true = 1, false = 0)) %>%
  # Dummy code the combined violent crimes categories, set user-missing values to 0.
  mutate(ACat12_Violent = if_else(ACat12 %in% c(2, 3, 7, 9, 12), true = 1, false = 0)) %>%
```

```
  # Dummy code the combined property crimes categories, set user-missing values to 0.
  mutate(ACat12_Property = if_else(ACat12 %in% c(1, 4, 8), true = 1, false = 0)) %>%
  # Dummy code the combined other crimes categories, set user-missing values to 0.
  mutate(ACat12_Other = if_else(ACat12 %in% c(6, 11), true = 1, false = 0)) %>%
  var_labels(ACat12_1   = "Arrested for arson",
             ACat12_2   = "Arrested for assault - DV, stalking",
             ACat12_3   = "Arrested for assault - non-sexual, non-DV",
             ACat12_4   = "Arrested for burglary",
             ACat12_5   = "Arrested for criminal sexual conduct",
             ACat12_6   = "Arrested for drug crime",
             ACat12_7   = "Arrested for homicide",
             ACat12_8   = "Arrested for larceny/theft/fraud",
             ACat12_9   = "Arrested for robbery",
             ACat12_10  = "Arrested for sex crime, other excluding CSC",
             ACat12_11  = "Arrested for traffic/ordinances",
             ACat12_12  = "Arrested for weapons",
             ACat12_999 = "Arrested for excluded user-missing",
             ACat12_Any = "Arrested for any of 12 main crime categories",
             ACat12_Sexual = "Arrested for any sexual crimes (2 categories)",
             ACat12_Violent = "Arrested for any violent crimes (5 categories)",
             ACat12_Property = "Arrested for any property crimes (3 categories)",
             ACat12_Other = "Arrested for any other crimes (2 categories)") ->
  ARR
```

```
read_sav(here::here("inst/extdata/SSA_CHG_PA_Charges_Imputed_2018-09-03.sav"),
         user_na = TRUE) %>%
  # Replace user-missing value labels.
  mutate(CCat12 = rvlabel(CCat12),
         CCat10 = rvlabel(CCat10),
         CCat04 = rvlabel(CCat04)) %>%
  # Dummy code crime category variables.
  dcode(x = ., y = factor(.$CCat12), stem = "CCat12") %>%
  # Dummy code any of the 12 main crime categories, set user-missing values to 0.
  mutate(CCat12_Any = if_else(CCat12 %in% 1:12, true = 1, false = 0)) %>%
  # Dummy code the combined sexual crimes categories, set user-missing values to 0.
  mutate(CCat12_Sexual = if_else(CCat12 %in% c(5, 10), true = 1, false = 0)) %>%
  # Dummy code the combined violent crimes categories, set user-missing values to 0.
  mutate(CCat12_Violent = if_else(CCat12 %in% c(2, 3, 7, 9, 12), true = 1, false = 0)) %>%
  # Dummy code the combined property crimes categories, set user-missing values to 0.
  mutate(CCat12_Property = if_else(CCat12 %in% c(1, 4, 8), true = 1, false = 0)) %>%
  # Dummy code the combined other crimes categories, set user-missing values to 0.
  mutate(CCat12_Other = if_else(CCat12 %in% c(6, 11), true = 1, false = 0)) %>%
  var_labels(CCat12_1   = "Charged for arson",
             CCat12_2   = "Charged for assault - DV, stalking",
             CCat12_3   = "Charged for assault - non-sexual, non-DV",
             CCat12_4   = "Charged for burglary",
             CCat12_5   = "Charged for criminal sexual conduct",
             CCat12_6   = "Charged for drug crime",
             CCat12_7   = "Charged for homicide",
             CCat12_8   = "Charged for larceny/theft/fraud",
             CCat12_9   = "Charged for robbery",
             CCat12_10  = "Charged for sex crime, other excluding CSC",
             CCat12_11  = "Charged for traffic/ordinances",
             CCat12_12  = "Charged for weapons",
             CCat12_999 = "Charged for excluded user-missing",
             CCat12_Any = "Charged for any of 12 main crime categories",
             CCat12_Sexual = "Charged for any sexual crimes (2 categories)",
             CCat12_Violent = "Charged for any violent crimes (5 categories)",
             CCat12_Property = "Charged for any property crimes (3 categories)",
             CCat12_Other = "Charged for any other crimes (2 categories)") ->
  CHG
```

```
read_sav(here::here("inst/extdata/SSA_JUD_Judicial_Charges_Imputed_2018-09-03.sav"),
         user_na = TRUE) %>%
  # Replace user-missing value labels.
  mutate(JCat12 = rvlabel(JCat12),
         JCat10 = rvlabel(JCat10),
         JCat04 = rvlabel(JCat04)) %>%
  # Dummy code crime category variables.
  dcode(x = ., y = factor(.$JCat12), stem = "JCat12") %>%
  # Dummy code any of the 12 main crime categories, set user-missing values to 0.
```

```
  mutate(JCat12_Any = if_else(JCat12 %in% 1:12, true = 1, false = 0)) %>%
  # Dummy code the combined sexual crimes categories, set user-missing values to 0.
  mutate(JCat12_Sexual = if_else(JCat12 %in% c(5, 10), true = 1, false = 0)) %>%
  # Dummy code the combined violent crimes categories, set user-missing values to 0.
  mutate(JCat12_Violent = if_else(JCat12 %in% c(2, 3, 7, 9, 12), true = 1, false = 0)) %>%
  # Dummy code the combined property crimes categories, set user-missing values to 0.
  mutate(JCat12_Property = if_else(JCat12 %in% c(1, 4, 8), true = 1, false = 0)) %>%
  # Dummy code the combined other crimes categories, set user-missing values to 0.
  mutate(JCat12_Other = if_else(JCat12 %in% c(6, 11), true = 1, false = 0)) %>%
  var_labels(JCat12_1   = "Adjudicated for arson",
             JCat12_2   = "Adjudicated for assault - DV, stalking",
             JCat12_3   = "Adjudicated for assault - non-sexual, non-DV",
             JCat12_4   = "Adjudicated for burglary",
             JCat12_5   = "Adjudicated for criminal sexual conduct",
             JCat12_6   = "Adjudicated for drug crime",
             JCat12_7   = "Adjudicated for homicide",
             JCat12_8   = "Adjudicated for larceny/theft/fraud",
             JCat12_9   = "Adjudicated for robbery",
             JCat12_10  = "Adjudicated for sex crime, other excluding CSC",
             JCat12_11  = "Adjudicated for traffic/ordinances",
             JCat12_12  = "Adjudicated for weapons",
             JCat12_999 = "Adjudicated for excluded user-missing",
             JCat12_Any = "Adjudicated for any of 12 main crime categories",
             JCat12_Sexual = "Adjudicated for any sexual crimes (2 categories)",
             JCat12_Violent = "Adjudicated for any violent crimes (5 categories)",
             JCat12_Property = "Adjudicated for any property crimes (3 categories)",
             JCat12_Other = "Adjudicated for any other crimes (2 categories)") ->
  JUD
```

5.2 SAK-Perpetrator Data File

We need this additional data file to identify the earliest SAK associated with each offender. See Campbell (2019) for details on this file.

```
# Read SPSS data file.
read_sav(here::here("inst/extdata/SAK_PERP_2018-04-03.sav"),
         user_na = TRUE) %>%
  # Sort by OID and SAK collection date (but NAs are last in sort order).
  arrange(OID, SDate.Yr, SDate.Mt, SDate) %>%
  # Add a variable showing how many unique SAKs the offender has.
  add_count(OID, name = "OID_NSAK") %>%
  # Make a OID variable label match the lbel used in CHR data sets.
  var_labels(OID = "Offender ID (de-identified)",
             OID_NSAK = "No. of unique SAKs associated with this offender") ->
  SPD
```

5.3 Scenarios for Alignment of Periods

The file read in below contains data used to show various potential scenarios for how the before, during, and after testing window periods can align with the observed portion of the criminal history records. These are *hypothetical data* manually constructed to yield a compact figure rather than examples from the actual data on specific offenders. The Scenarios data frame is designed to be used with the *vistime::gg_vistime()* function.

```
Scenarios <- read.csv(file = here::here("inst/extdata/TWindow_Scenarios.csv"),
                      sep = "\t")
```

6 Data Management

6.1 Create Conviction Dataset

Convictions are the subset of adjudicated charges that have a specific disposition code indicating that the offender was convicted. For simplicity, we extract them to a new tibble. Note that in this *CON* tibble, the *NCON* variable is marking all

convictions, even those for crimes that fall into the user-missing category *JCat12* = 999.

```
# Create a subset of JUD containing only charges where offender was convicted.
JUD %>%
  # Retain a subset of JUD records based on disposition.
  filter(JDispCat == 1) %>%
  # Add a variable to CON to simplify aggregation later.
  mutate(NCON = 1) %>%
  # Add variable labels.
  var_labels(JCat12_1   = "Convicted for arson",
             JCat12_2   = "Convicted for assault - DV, stalking",
             JCat12_3   = "Convicted for assault - non-sexual, non-DV",
             JCat12_4   = "Convicted for burglary",
             JCat12_5   = "Convicted for criminal sexual conduct",
             JCat12_6   = "Convicted for drug crime",
             JCat12_7   = "Convicted for homicide",
             JCat12_8   = "Convicted for larceny/theft/fraud",
             JCat12_9   = "Convicted for robbery",
             JCat12_10  = "Convicted for sex crime, other excluding CSC",
             JCat12_11  = "Convicted for traffic/ordinances",
             JCat12_12  = "Convicted for weapons",
             JCat12_999 = "Convicted for excluded user-missing",
             JCat12_Any = "Convicted for any of 12 main crime categories",
             JCat12_Sexual = "Convicted for any sexual crimes (2 categories)",
             JCat12_Violent = "Convicted for any violent crimes (5 categories)",
             JCat12_Property = "Convicted for any property crimes (3 categories)",
             JCat12_Other = "Convicted for any other crimes (2 categories)",
             NCON       = "Convicted charge record w/ conviction") ->
  CON
```

6.2 Merge Number of Convictions onto IDN Dataset

```
# Merge NCON conviction charge count variable into IDN.
CON %>%
  group_by(OID) %>%
  summarize_at(.vars = "NCON", .funs = c("sum")) %>%
  ungroup() %>%
  left_join(x = IDN, y = ., by = "OID") %>%
  # Recode NCON variable to replace NA with 0.
  replace_na(data = ., replace = list(NCON = 0)) %>%
  # Retain only variables we need, in sensible order.
  select(all_of(c(names(IDN), "NCON"))) %>%
  # Add variable labels.
  var_labels(NCON = "No. of adjudicated charge records w/ convictions") ->
  IDN
```

6.3 Identify Earliest SAK for Each Offender

For the paper, we need to identify the earliest SAK associated with each offender who has criminal history data and store those in a new dataset. Along the way, we need to do several intermediate tasks.

According to the help files, *dplyr::arrange* always puts NA values last in the sort order. That means we need to use a trick to adjust for that when we use that function to sort the data.

First, we have to trim the SPD data down to just the SAKs associated with offenders for whom we have criminal history data (SPDCHR).

```
SPD %>%
  # Retain only offenders with CHR data.
  filter(OIDinCHR == "Yes") ->
  SPDCHR
```

With the resulting *SPDCHR* data, we need to identify the start of the testing window for each SAK and store it in a new variable called *WDate*, then aggregate across SAKs associated with each offender to identify the earliest *WDate* value.

6.3.1 Identify Start of Testing Window for each SAK

The testing window for a SAK should start on the date it was collected. We already have a variable called *SDate* that records this information. However, some SAKs have only partial date information, leaving *SDate* with NA values. A very small number of SAKs have completely missing data on *SDate* such that not even the year of collection is known, but most of the cases with missing *SDate* values have the year of collection recorded in *SDate.Yr* and some also have the month recorded in *SDate.Mt*.

SAK collection dates (*SDate*) were recorded by hospitals (or other healthcare providers) who actually collected the SAKs. These dates are very likely to be accurate because healthcare providers have a strong financial incentive to get dates of service correct for billing purposes.

However, when we have missing or partial *SDate* data, we have an additional source of data that might provide an exact date for the start of the testing window (*WDate*). In our prior work with these data, we linked some SAKs to existing CHR incident records that involved criminal sexual conduct (as evidenced by an associated arrest offense record, prosecutor charge record, or judicial charge record for a sexual assault). Those links were created to recognize possible overlap between sexual assaults represented in the CODIS hits (CHITS) data from the SAK testing and the CHR data files. We did not require exact date matches in the previous work because we were dealing with data that originated from different organizations. We allowed up to a 45 day discrepancy in the dates when making matches.

For SAKs that are not linked to an incident record, *OID_IID* = " " (an empty string) and *OID_IDate* = NA. Therefore, the only date information available is in the SAK collection date variables (*SDate*, *SDate.Yr*, and *SDate.Mt*). If we have a valid value in *SDate*, then *WDate* = *SDate*. When we only have partial date information, then *SDate* is missing but *SDate.Yr* and possibly also *SDate.Mt* have valid values, but we cannot assign a precise date to the SAK collection and *WDate* ends up with a missing value.

For SAKs that were linked to an incident record, *OID_IID* stores the incident ID value that links to the CHR incident record in the *INC* data and *OID_IDate* stores the corresponding incident date. Thus, if *SDate* is missing, we set *WDate* = *OID_IDate* instead, thereby using *OID_IDate* as supplementary information to reduce missing data in *WDate*.

We also create a *TWindow* date interval variable to record the start and end dates for the 118-day testing window for each of these SAKs.

```
SPDCHR %>%
  mutate(# Testing Window Start Date.
         WDate = case_when(
           # Default to using SDate if it is not missing.
           is.na(SDate) == FALSE ~ SDate,
           # If SDate is incomplete or missing, use a valid OID_IDate instead.
           is.na(SDate) == TRUE & is.na(OID_IDate) == FALSE ~ OID_IDate),
         # Testing Window Start Year.
         WDate.Yr = case_when(
           # Default to year of WDate if it is not missing.
           is.na(WDate) == FALSE ~ year(WDate),
           # Otherwise use SDate.Yr
           is.na(WDate) == TRUE ~ SDate.Yr),
         # Testing Window Start Month.
         WDate.Mt = case_when(
           is.na(WDate) == FALSE ~ month(WDate),
           is.na(WDate) == TRUE ~ SDate.Mt),
         # Testing Window Start Date Status.
         WDate.Status = case_when(
           is.na(WDate) == TRUE & is.na(WDate.Yr) == TRUE ~ "Missing",
           is.na(WDate) == TRUE & is.na(WDate.Mt) == TRUE ~ "Partial.Y",
           is.na(WDate) == TRUE & is.na(WDate.Mt) == FALSE ~ "Partial.YM",
           is.na(WDate) == FALSE ~ "Known"),
         # Testing Window Date Interval.
         TWindow = interval(WDate, WDate + 118)) %>%
  # Sort records by year, month, and date but ensure that within each layer
  # records with missing data come first.
  arrange(OID, -is.na(WDate.Yr), WDate.Yr, -is.na(WDate.Mt), WDate.Mt, WDate) %>%
  # Group by OID, then select first record for each offender.
  group_by(OID) %>%
  filter(row_number() == 1) %>%
  # Ungroup to ensure simpler, more predictable tibble behavior later.
  ungroup() %>%
  # Add variable labels.
```

```
  var_labels(WDate = "Testing Window Start Date",
             WDate.Yr = "Testing Window Start Year",
             WDate.Mt = "Testing Window Start Month",
             WDate.Status = "Testing Window Start Date Status",
             TWindow = "Testing Window Date Interval") ->
  SPDCHRE
```

The code above takes the *SPDCHR* dataset, adds new variables pertaining to the start of the testing window, sorts the data into chronological order, then drops everything but the earliest SAK for each perpetrator. The result is saved to the *SPDCHRE* dataset

6.3.2 Missingness Status for Earliest SAK Testing Window Start Date

Here we pause to assess how many of the offenders with CHR data have missing *WDate* values for their earliest SAK. Table 1 shows a cross-tabulation of the number of SAKs associated with a perpetrator (*OID_NSAK*) against the status of the *WDate* variable.

```
# Table caption.
TCap <- paste("Number of SAKs by WDate Status for Perpetrator's Earliest SAK")
# Footnote text.
FN <- paste("Cells contain counts of perpetrators. Only perpetrators for whom",
            "criminal history records are available were included.",
            "No. SAKs, total number of SAKs associated with perpetrator;",
            "WDate, start date for testing window;",
            "Y, M, and D, respectively refer to the year, month, and day",
            "components of WDate. Letters replaced by question marks (?)",
            "show which components are unknown.")

# Vector of text values for column labels to be used in the table.
clabels <- c("No. SAKs", "YMD", "YM?", "Y??", "???", "Sum")

addmargins(xtabs(~OID_NSAK + WDate.Status, addNA = TRUE, data = SPDCHRE)) %>%
  as.data.frame() %>%
  pivot_wider(names_from = WDate.Status, values_from = Freq) %>%
  select(OID_NSAK, Known, Partial.YM, Partial.Y, Missing, Sum) %>%
  kable(., format = "latex", booktabs = TRUE, caption = TCap,
        col.names = clabels, format.args = list(big.mark = ",")) %>%
  add_header_above(c("", "Known", "Partial" = 2, "Missing", "")) %>%
  add_header_above(c("", "WDate Status" = 4, "")) %>%
  column_spec(1, width = "1.8cm") %>%
  column_spec(2:6, width = "1.2cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

```
SPDCHRE %>%
  # Retain only the earliest SAKs with known WDate values.
  filter(WDate.Status == "Known") %>%
  # Rename variables for clarity.
  dplyr::rename(ESAKID = SAKID,
                ESAK_IID = OID_IID) %>%
  # Add variable labels.
  var_labels(ESAKID = "SAK ID for Offender's Earliest SAK",
             ESAK_IID = "Incident ID for Offender's Earliest SAK",
             TWindow = "Testing Window Date Interval") ->
  SPDCHREW
```

We have complete, known *WDate* values for 1082 (95%) of the 1142 perpetrators with CHR data. That leaves 60 perpetrators with missing or partial information for *WDate*. For perpetrators completely missing values on one or more of their SAKs, it is impossible to tell which was the earliest SAK. There is no way we can split their criminal histories into the three periods for before, during, and after the testing window.

We know only the year of earliest SAK collection for most of the perpetrators with partial *WDate* information. We have not identified a reasonable and defensible method for imputing precise *WDate* values for these offenders. Therefore, we have excluded them from the sample analyzed for this paper by trimming the *SPDCHRE* dataset down to just those offenders who have a known *WDate* and saving the results as the *SPDCHREW* dataset. This is an offender-level dataset.

	WDate Status				
	Known	Partial		Missing	
No. SAKs	YMD	YM?	Y??	???	Sum
1	985	1	46	1	1,033
2	64	0	4	2	70
3	16	0	2	0	18
4	7	0	2	0	9
5	3	0	1	0	4
6	2	0	1	0	3
7	1	0	0	0	1
8	1	0	0	0	1
9	2	0	0	0	2
10	1	0	0	0	1
Sum	1,082	1	56	3	1,142

Note: Cells contain counts of perpetrators. Only perpetrators for whom criminal history records are available were included. No. SAKs, total number of SAKs associated with perpetrator; WDate, start date for testing window; Y, M, and D, respectively refer to the year, month, and day components of WDate. Letters replaced by question marks (?) show which components are unknown.

Table 1: Number of SAKs by WDate Status for Perpetrator's Earliest SAK

6.4 Merge Testing Window Variables Onto IDN Data

Merging the testing window variables onto the *IDN* data is an obvious next step. Then we subset down to the records that have non-missing *WDate* values and select only the variables we need to keep before setting a variable label and saving to a new tibble called *IDNEW*.

After merging in *TWindow*, it is now possible to compute date intervals for the periods of the recorded adult criminal history observed before and after the testing window. We call those new variables *BWindow* and *AWindow* and supplement all three date intervals with duration variables measuring the duration of the before, during, and after periods in years (*Years_Before*, *Years_During*, and *Years_After*, respectively). We will eventually use these duration variables as offsets when modeling count outcome variables that need to adjust for period duration to produce crime incidence rate variables.

We are working with adult criminal history records that only include data for crimes committed since each offender reached 16 years of age. We create a date interval variable called *OWindow* that starts on the offender's 16th birthday (*Age16Date*, which can be computed from date of birth *DOB*) and ends on the data collection date when the criminal history data were extracted from the state data warehouse: April 15, 2016 (*DCDate*). The duration (in years) of that overall observed adult CHR period is stored in *Years_Overall*.

One might assume that the offender's 16th birth day (*Age16Date*) should be the start date for the before period. However, because *WDate* is derived from when the earliest SAK for the offender was collected rather than the criminal history data, *WDate* can in principle either precede or occur on *Age16Date*.

Figure 1 shows a set of hypothetical scenarios that could occur and how much of the before, during, and after testing window periods are actually observed.

```
# Figure caption.
FCap <- paste("Scenarios for Alignment of Before, During, and After Testing",
              "Window Periods with Observed Adult Criminal History Records",
              "Period.",
              "These are based on hypothetical data for a cohort of offenders",
              "born on 1998-04-16 with CHR data collected on 2016-04-15.",
              "Age 16, date of offender's 16th birthday;",
              "CHR, criminal history records; SAK, sexual assault kit. ")
```

```
Scenarios %>%
  filter(type == "event") %>%
  gg_vistime(., optimize_y = FALSE, col.group = "scenario", linewidth = 4) ->
  F1

# Plot with annotations
F1 + theme(text = element_text(size = 15, color = "black")) +
  annotate(geom = "text", x = as_datetime("2013-06-30"), y = 65, size = 3,
             label = "Age16Date < (WDate - 1)", hjust = 0.21) +
  annotate(geom = "text", x = as_datetime("2013-06-30"), y = 53, size = 3,
             label = "Age16Date = (WDate - 1)", hjust = 0.21) +
  annotate(geom = "text", x = as_datetime("2013-06-30"), y = 49.25, size = 3,
             label = "1 day observed", hjust = 0) +
  annotate(geom = "text", x = as_datetime("2013-06-30"), y = 44, size = 3,
             label = "Age16Date = WDate", hjust = 0.25) +
  annotate(geom = "text", x = as_datetime("2013-06-30"), y = 40.25, size = 3,
             label = "0 days observed", hjust = 0) +
  annotate(geom = "text", x = as_datetime("2013-06-30"), y = 35, size = 3,
             label = "WDate < Age16Date < (WDate + 118)", hjust = .14) +
  annotate(geom = "text", x = as_datetime("2013-06-30"), y = 26, size = 3,
             label = "Age16Date = (WDate + 118)", hjust = 0.19) +
  annotate(geom = "text", x = as_datetime("2013-12-19"), y = 21, size = 3,
             label = "1 day observed", hjust = 0) +
  annotate(geom = "text", x = as_datetime("2013-06-30"), y = 17, size = 3,
             label = "Age16Date = (WDate + 119)", hjust = 0.19) +
  annotate(geom = "text", x = as_datetime("2013-12-19"), y = 12, size = 3,
             label = "0 days observed", hjust = 0) +
  annotate(geom = "text", x = as_datetime("2014-04-16"), y = 10, size = 3,
             label = "All days observed", hjust = 0) +
  annotate(geom = "text", x = as_datetime("2013-06-30"), y = 8, size = 3,
             label = "Age16Date > (WDate + 119)", hjust = 0.19) +
  annotate(geom = "text", x = as_datetime("2014-02-14"), y = 1, size = 3,
             label = "Most but not all days observed", hjust = 0)
```

We create variables below that help us determine how often each scenario occurred and adjust the before period date interval in a sensible way when it does. We defined the before period as starting on the offender's 16th birthday (*Age16Date*) and ending one day before the earliest SAK was collected (*WDate*) when $Age16Date < WDate$, and otherwise as both starting and ending the day before *WDate*. That yields a before period with a positive duration for most offenders but zero years for those whose earliest SAK was collected before they reached age 16 years.

For all offenders, the after period starts on the 119th day after *WDate* (because the testing window is 118 days long) and ends on the CHR data collection date (*DCDate*).

The chunk below was used interactively to select a color palette for Figure 1 that is good for visual discrimination of qualitative categories that are not ordered. We wanted colors light enough to have adequate contrast with black text. Using the Paired color palette to encode whether a period was unobserved versus observed by intensity of the hue ended up looking too cluttered and contrast was poor, so we used selected colors from Set2. The chunk is disabled now to reduce clutter in the output. The colors assigned for events and fonts are stored as columns in the Scenarios data frame.

```
# Show a range of color palettes that might work.
RColorBrewer::display.brewer.all(type = "qual")

# Show hexadecimal codes for colors from specific palettes.
RColorBrewer::brewer.pal(8, "Set2")
#[1] "#66C2A5" "#FC8D62" "#8DA0CB" "#E78AC3" "#A6D854" "#FFD92F" "#E5C494" "#B3B3B3"
RColorBrewer::brewer.pal(12, "Paired")
#[1] "#A6CEE3" "#1F78B4" "#B2DF8A" "#33A02C" "#FB9A99" "#E31A1C" "#FDBF6F" "#FF7F00"
#[9] "#CAB2D6" "#6A3D9A" "#FFFF99" "#B15928"
```

The chunk below merges variables from SPDCHREW into IDN, then creates a number of variables as described above in this section.

```
SPDCHREW %>%
  # Retain only variables we need to merge into IDN.
  select(ESAKID, OID, ESAK_IID, WDate, TWindow) %>%
  # Merge SPDCHREW variables onto INC rows.
  right_join(x = ., y = IDN, by = "OID") %>%
```

```
  # Retain only IDN records for offenders with non-missing WDate values.
  filter(is.na(WDate) == FALSE) %>%
  # Create more variables.
  mutate(
    # Offender 16th birthday (start of before period).
    Age16Date = DOB + dyears(16),
    # Offender age at start of earliest SAK testing window
    AgeWDate = interval(DOB, WDate)/dyears(1),
    # Scenario for alignment of periods.
    Scenario = case_when(
      Age16Date < WDate - 1 ~ "A",
      Age16Date == WDate - 1 ~ "B",
      Age16Date == WDate ~ "C",
      WDate < Age16Date & Age16Date < WDate + 118 ~ "D",
      Age16Date == WDate + 118 ~ "E",
      Age16Date == WDate + 119 ~ "F",
      Age16Date > WDate + 118 ~ "G"),
    # Convert to a factor for later convenience.
    Scenario = factor(Scenario, levels = c("A", "B", "C", "D", "E", "F", "G")),
    # Did offender turn 16 before WDate (start of testing window, TWindow)?
    Turn16_Before_WDate = (Age16Date < WDate),
    # Did offender turn 16 on or after WDate (start of testing window, TWindow)?
    Turn16_After_WDate = (Age16Date >= WDate),
    # Data Collection Date (end of after period)
    DCDate = ymd("2016-04-15", tz = "UTC"),
    # Did testing window end before data collection date?
    End_TWindow_Before_DC = (WDate + 118 < DCDate),
    # Observed window date interval
    OWindow = interval(Age16Date, DCDate),
    # Before testing window date interval
    BWindow = case_when(
      # If offender turned 16 before WDate, before period lasts > 0 years.
      Turn16_Before_WDate ~ interval(Age16Date, WDate - 1),
      # If offender turned 16 on or after WDate, before period lasts 0 years.
      Turn16_After_WDate ~ interval(WDate - 1, WDate - 1)),
    # After testing window date interval.
    AWindow = interval(WDate + 119, DCDate),
    # Duration of observed period (from age 16 to data collection date)
    Years_Overall = (as.duration(OWindow) + ddays(1))/dyears(1),
    # Duration of before testing window period (years).
    Years_Before = case_when(
      Age16Date <  WDate - 1 ~ (as.duration(BWindow) + ddays(1))/dyears(1),
      Age16Date == WDate - 1 ~ ddays(1)/dyears(1),
      Age16Date >  WDate - 1 ~ 0),
    # Duration of during testing window period (years)
    Years_During = case_when(
      Age16Date < WDate ~ ddays(119)/dyears(1),
      Age16Date %within% TWindow ~
        (as.duration(intersect(OWindow, TWindow)) + ddays(1))/dyears(1),
      Age16Date > WDate + 118 ~ 0),
    # Duration of after testing window period (years)
    Years_After = case_when(
      Age16Date <= WDate + 119 ~ (as.duration(AWindow) + ddays(1))/dyears(1),
      Age16Date >  WDate + 119 ~
        (as.duration(intersect(OWindow, AWindow)) + ddays(1))/dyears(1))) %>%
  var_labels(
    Age16Date = "Offender 16th Birthday",
    AgeWDate = "Offender Age at Start of Earliest SAK Testing Window (Years)",
    Scenario = "Scenario for alignment of before, during, and after testing window periods with observed criminal history records period
    Turn16_Before_WDate = "Offender Turned 16 Before Start of Testing Window",
    Turn16_After_WDate = "Offender Turned 16 On or After Start of Testing Window",
    DCDate = "Data Collection Date for Criminal history Records",
    End_TWindow_Before_DC = "Testing Window Ended Before Data Collection Date",
    Years_Overall = "Duration of observed period (years)",
    Years_Before = "Duration of before testing window period (years)",
    Years_During = "Duration of during testing window period (years)",
    Years_After = "Duration of after testing window period (years)",
    BWindow = "Before Testing Window Date Interval",
    TWindow = "Testing Window Date Interval",
    AWindow = "After Testing Window Date Interval") %>%
  # Retain only variables we need, in sensible order.
```

```
  select(all_of(c(names(IDN), "Age16Date", "AgeWDate", "Scenario", "ESAKID",
                  "ESAK_IID", "WDate", "TWindow", "Turn16_Before_WDate",
                  "Turn16_After_WDate", "DCDate", "End_TWindow_Before_DC",
                  "OWindow", "BWindow", "AWindow", "Years_Overall",
                  "Years_Before", "Years_During", "Years_After"))) ->
  IDNEW
```

We end up with an *IDNEW* tibble that has 1082 of the original 1142 perpetrators with CHR data. Only 41 of the 1082 offenders turned 16 on or after their earliest SAK was collected and the testing window had started; they therefore have *Years_Before* = 0 years. Table 2 shows the frequency distribution for scenarios actually observed among the offenders.

```
# Table caption.
TCap <- paste("Frequency Distribution for Scenario of Alignment of Before,",
              "During, and After Testing Window Periods with Observed",
              "Criminal History Records Period")
# Footnote text.
FN <- paste("Offsets from specific dates (e.g., - 1) mentioned in criteria",
            "are in units of days.")

# Gather scenario descriptions to merge int data from offenders.
Scenarios %>%
  filter(type == "comment") %>%
  select(scenario, event, criteria) %>%
  rename(Scenario = scenario, Description = event, Criteria = criteria) ->
  ScenarioTable

IDNEW %>%
  select(OID, Scenario) %>%
  group_by(Scenario, .drop = FALSE) %>%
  count() %>%
  rename(Frequency = n) %>%
  mutate(Percent = 100*Frequency/nrow(IDNEW)) %>%
  # Merge in scenario descriptions.
  left_join(x = ScenarioTable, y = ., by = "Scenario") %>%
  kable(format = "latex", booktabs = TRUE, digits = 2,
        caption = TCap) %>%
  kable_styling() %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

Scenario	Description	Criteria	Frequency	Percent
A	Before (all), During (all), After (all)	Age16Date < WDate - 1	1040	96.12
B	Before (1 day), During (all), After (all)	Age16Date = WDate - 1	1	0.09
C	Before (none), During (all), After (all)	Age16Date = WDate	0	0.00
D	Before (none), During (part), After (all)	WDate < Age16Date < WDate + 118	11	1.02
E	Before (none), During (1 day), After (all)	Age16Date = WDate + 118	0	0.00
F	Before (none), During (none), After (all)	Age16Date = WDate + 119	0	0.00
G	Before (none), During (none), After (part)	Age16Date > WDate + 119	30	2.77

Note: Offsets from specific dates (e.g., - 1) mentioned in criteria are in units of days.

Table 2: Frequency Distribution for Scenario of Alignment of Before, During, and After Testing Window Periods with Observed Criminal History Records Period

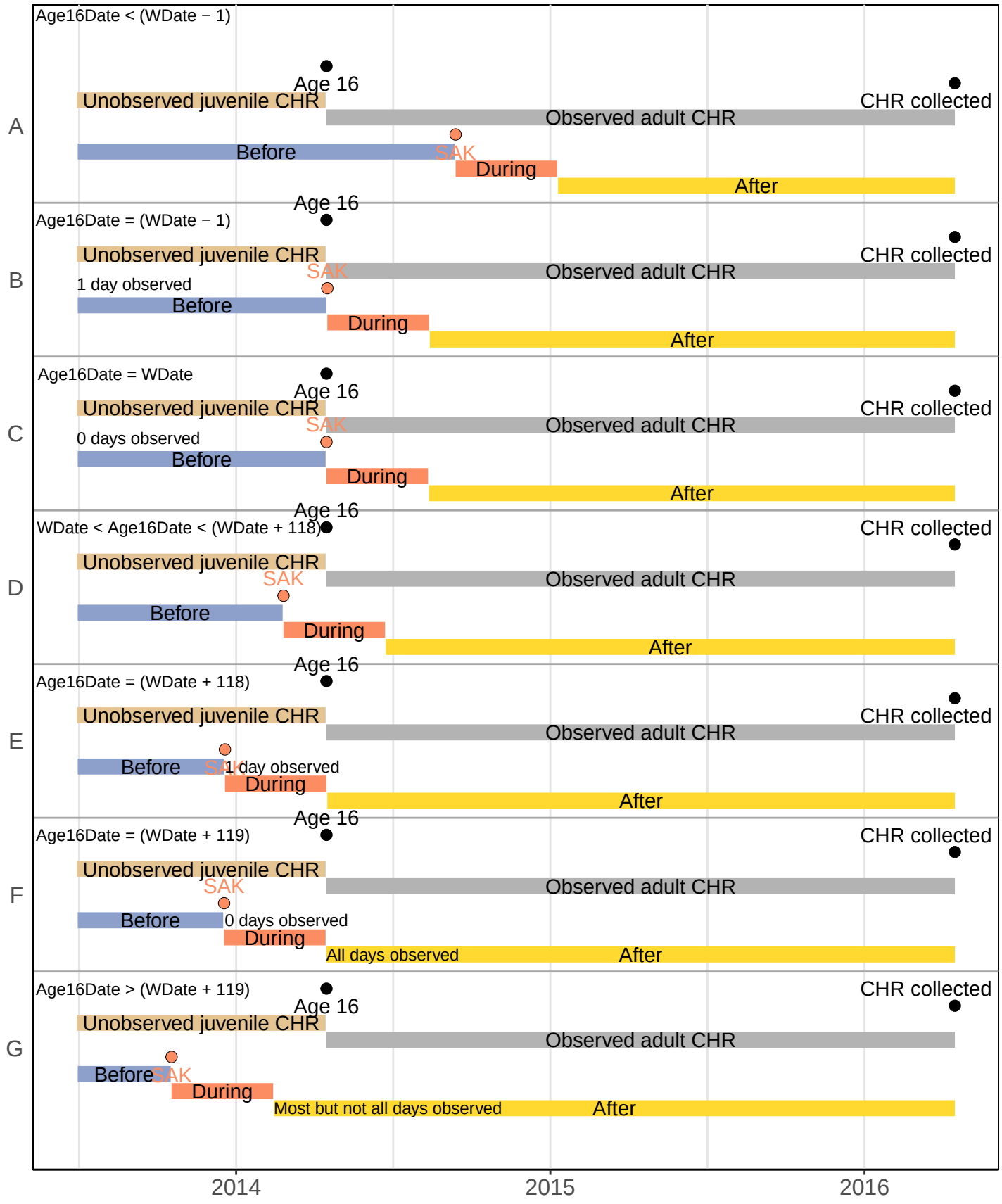

Figure 1: Scenarios for Alignment of Before, During, and After Testing Window Periods with Observed Adult Criminal History Records Period. These are based on hypothetical data for a cohort of offenders born on 1998-04-16 with CHR data collected on 2016-04-15. Age 16, date of offender's 16th birthday; CHR, criminal history records; SAK, sexual assault kit.

6.5 Merge Testing Window Variables Onto Incident Data

The next step is to merge variables from *IDNEW* onto the incident data in *INC* so that we can classify each incident according to whether it occurred before, during, or after the earliest testing window for the offender in the new *IWhen* variable. Along the way we rename some variables and update variable labels for clarity. Then we select only the variables we need to keep and subset down to the records that have non-missing *WDate* values and save to a new tibble called *INCEW*.

The *INCEW* tibble still contains one record for every incident associated with the offenders contained in *IDNEW*, regardless of what crime categories were associated with the incident. This will change later in the script, after we construct and merge in additional variables that make it easy to subset *INCEW* to records for incidents associated with at least one arrest, charge, or conviction for one of the 12 main crime categories of interest.

```
# Object to hold names of IWhen levels.
IWhenLevels <- c("Before", "During", "After")

# Object to hold names of crucial testing window variables
twvars <- c("ESAKID", "ESAK_IID", "WDate", "TWindow", "IWhen")

# Data management for the merge.
IDNEW %>%
  # Retain only variables we need to merge into INC.
  select(ESAKID, OID, ESAK_IID, WDate, TWindow) %>%
  # Merge IDNEW variables onto INC rows.
  right_join(x = ., y = INC, by = "OID") %>%
  # Retain only INC records for offenders with non-missing WDate values.
  filter(is.na(WDate) == FALSE) %>%
  # Add more variables.
  mutate(# Classify incidents as before, during, or after the testing window.
         IWhen = case_when(
           # If incident matched to earliest SAK, incident was in the window.
           ESAK_IID == IID ~ "During",
           # Otherwise, compare IDate to TWindow to determine if incident was in
           # the window.
           IDate < int_start(TWindow) ~ "Before",
           IDate %within% TWindow ~ "During",
           IDate > int_end(TWindow) ~ "After")) %>%
  # Convert IWhen to a factor with levels in sensible order for display.
  mutate(IWhen = factor(IWhen, levels = IWhenLevels, ordered = TRUE)) %>%
  # Create IYearsAfterTW to store number of years between IDate and end of
  # testing window.
  mutate(IYearsAfterTW = (as.duration(interval(int_end(TWindow), IDate)) +
                          ddays(1))/dyears(1)) %>%
  # Add variable labels.
  var_labels(IWhen = "When was incident relative to testing window?") %>%
  # Retain only variables we need, in sensible order.
  select(all_of(c(names(INC), twvars, "IYearsAfterTW"))) %>%
  # Add another variable label that was otherwise being dropped.
  var_labels(TWindow = "Testing Window Date Interval",
             IYearsAfterTW = "No. Years Incident Occurred After End of Testing Window") ->
  INCEW
```

Table 3 shows the frequency distribution for when the incidents occurred relative to the testing window.

```
# Table caption.
TCap <- paste("Frequency Distribution of When Incident Occurred Relative to",
              "Testing Window")
# Footnote text.
FN <- paste("Only incidents with valid testing window start dates were",
            "included. IWhen, when was incident relative to testing window?")

# Vector of text values for column labels to be used in the table.
clabels <- c("IWhen", "Frequency", "Percent")

xtabs(~IWhen, data = INCEW) %>%
  addmargins() %>%
  as.data.frame() %>%
  mutate(Percent = 100*Freq/nrow(INCEW)) %>%
  kable(., format = "latex", booktabs = TRUE, caption = TCap, digits = 2,
        col.names = clabels, format.args = list(big.mark = ",")) %>%
```

```
  column_spec(1:3, width = "1.5cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

IWhen	Frequency	Percent
Before	2,652	29.22
During	423	4.66
After	6,002	66.12
Sum	9,077	100.00

Note: Only incidents with valid testing window start dates were included. IWhen, when was incident relative to testing window?

Table 3: Frequency Distribution of When Incident Occurred Relative to Testing Window

We used one special case decision rule when classifying incidents. If an incident record had been previously linked to the perpetrator's earliest SAK in the SPD tibble, then the *IID* variable in the INC record will have a value matching the *ESAK_IID* variable merged in from the IDNEW tibble. Such incidents must by definition be treated as occurring during the testing window even if the *IDate* lies outside the window. This reflects the notion that we are trusting the *SDate* values more than *IDate* values when constructing *WDate* and *TWindow*. Table 4 shows that there are only a few incidents matched to the perpetrator's earliest SAK where this decision rule classifies an incident as occurring during the testing window even though *IDate* actually falls outside that date interval.

```
# Table caption.
TCap <- paste("Crosstabulation of Whether Incident Date Was Within Testing",
              "Window and Whether Incident Matched to Perpetrator's Earliest",
              "SAK")
# Footnote text.
FN <- paste("Cells contain counts of incidents. Only incidents with valid",
            "testing window start dates were included. ESAK, earliest sexual",
            "assault kit; IDate, incident date; TWindow, testing window.")

# Vector of text values for column labels to be used in the table.
clabels <- c("IDate in TWindow", "FALSE", "TRUE", "Sum")

xtabs(~IDate %within% TWindow + (ESAK_IID == IID), addNA = TRUE, data = INCEW) %>%
  addmargins() %>%
  as.data.frame() %>%
  pivot_wider(names_from = ESAK_IID....IID, values_from = Freq) %>%
  kable(., format = "latex", booktabs = TRUE, caption = TCap,
        col.names = clabels, format.args = list(big.mark = ",")) %>%
  add_header_above(c("", "Matched to ESAK" = 2, "")) %>%
  column_spec(1, width = "3cm") %>%
  column_spec(2:4, width = "1.5cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

Now, we need to get offender-level counts of incidents that occurred before, during, and after the testing window respectively called *NINC_Before*, *NINC_During*, and *NINC_After*. We can aggregate as follows to get them. We also merge these variables into *IDNEW*. However, these incident count variables are not restricted to incidents involving arrest, charge, or conviction for one or more of the 12 main crime categories. We will construct additional variables consistent with that criterion later.

```
INCEW %>%
  mutate(NINC = 1) %>%
  group_by(OID, IWhen, .drop = FALSE) %>%
  # Aggregate by OID and IWhen to get count variables
  summarize(NINC = sum(NINC)) %>%
  pivot_wider(names_from = IWhen, values_from = NINC,
              names_glue = "NINC_{IWhen}") %>%
  var_labels(NINC_Before = "No. of incident records before testing window",
```

	Matched to ESAK		
IDate in TWindow	FALSE	TRUE	Sum
FALSE	8,654	15	8,669
TRUE	257	151	408
Sum	8,911	166	9,077

Note: Cells contain counts of incidents. Only incidents with valid testing window start dates were included. ESAK, earliest sexual assault kit; IDate, incident date; TWindow, testing window.

Table 4: Crosstabulation of Whether Incident Date Was Within Testing Window and Whether Incident Matched to Perpetrator's Earliest SAK

```
            NINC_During = "No. of incident records during testing window",
            NINC_After  = "No. of incident records after testing window") %>%
  # Merge incident record count variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(c("NINC_Before", "NINC_During", "NINC_After")),
                .fns = replace_na, replace = 0)) %>%
  # Set count variables with values of 0 to NA when that period was unobserved.
  mutate(across(.cols = c("NINC_Before"),
                .fns = ~ifelse(Years_Before == 0, yes = NA, no = .))) %>%
  mutate(across(.cols = c("NINC_During"),
                .fns = ~ifelse(Years_During == 0, yes = NA, no = .))) ->
  IDNEW
```

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

6.6 Merge Testing Window Variables Onto Arrest Records

The next step is to merge variables from *INCEW* onto the arrest data in *ARR* so that we can classify each arrest offense record according to whether the incident associated with it occurred before, during, or after the earliest testing window for the offender by using *IWhen*. Then we subset down to the *ARR* records that have non-missing *WDate* values plus *ACat12_Any* = 1, then select only the variables we need to keep before saving a new tibble called *ARREW*.

```
INCEW %>%
  # Retain only variables we need to merge into ARR.
  select(IID, ESAKID, ESAK_IID, WDate, TWindow, IWhen) %>%
  # Merge INCEW variables onto ARR rows.
  right_join(x = ., y = ARR, by = "IID") %>%
  # Retain only ARR records for offenders with non-missing WDate values.
  filter(is.na(WDate) == FALSE) %>%
  # Retain only ARR records for the main 12 crime categories, dropping 999.
  filter(ACat12_Any == 1) %>%
  # Retain only variables we need, in sensible order.
  select(all_of(c(names(ARR), twvars))) %>%
  # Add a variable label that was otherwise being dropped.
  var_labels(TWindow = "Testing Window Date Interval") ->
  ARREW
```

The original *ARR* tibble had 9826 arrest offense records. Dropping the records associated with offenders whose earliest testing window could not be identified or that were coded *ACat12_Any* = 0 because *ACat12* = 999 (excluded user-missing) leaves 8945 arrest offense records in *ARREW*. Table 5 shows how many arrest offense records were associated with incidents that occurred before, during, and after the testing window.

```
# Table caption.
TCap <- paste("Frequency Distribution of When the Incident Associated With An",
              "Arrest Offense Record Occurred Relative to Testing Window")
# Footnote text.
```

```
FN <- paste("Only arrest offense records for incidents with valid testing",
            "window start dates and offenses from the 12 crime categories",
            "of interest were included. IWhen, when was incident relative to",
            "testing window?")

# Vector of text values for column labels to be used in the table.
clabels <- c("IWhen", "Frequency", "Percent")

xtabs(~IWhen, data = ARREW) %>%
  addmargins() %>%
  as.data.frame() %>%
  mutate(Percent = 100*Freq/nrow(ARREW)) %>%
  kable(., format = "latex", booktabs = TRUE, caption = TCap, digits = 2,
        col.names = clabels, format.args = list(big.mark = ",")) %>%
  column_spec(1:3, width = "1.5cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

IWhen	Frequency	Percent
Before	2,642	29.54
During	419	4.68
After	5,884	65.78
Sum	8,945	100.00

Note:
Only arrest offense records for incidents with valid testing window start dates and offenses from the 12 crime categories of interest were included. IWhen, when was incident relative to testing window?

Table 5: Frequency Distribution of When the Incident Associated With An Arrest Offense Record Occurred Relative to Testing Window

Now, we need to get offender-level counts of arrest offense records associated with incidents that occurred before, during, and after the testing window respectively called *NARR12_Before*, *NARR12_During*, and *NARR12_After*. We can aggregate as follows to get them. We merge these variables into *IDNEW* too. These new aggregated variables respect the criterion of counting only offenses in the 12 main crime categories of interest because of the case selection that went into building the *ARREW* data.

```
ARREW %>%
  mutate(NARR12 = 1) %>%
  group_by(OID, IWhen, .drop = FALSE) %>%
  # Aggregate by OID and IWhen to get count variables
  summarize(NARR12 = sum(NARR12)) %>%
  pivot_wider(names_from = IWhen, values_from = NARR12,
              names_glue = "NARR12_{IWhen}") %>%
  var_labels(NARR12_Before = "No. of arrest offense records from 12 main crime categories before testing window",
             NARR12_During = "No. of arrest offense records from 12 main crime categories during testing window",
             NARR12_After  = "No. of arrest offense records from 12 main crime categories after testing window")  %>%
  # Merge arrest offense record count variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(c("NARR12_Before", "NARR12_During", "NARR12_After")),
                .fns = replace_na, replace = 0)) %>%
  # Compute alternate NARR for records from 12 main crime categories.
  mutate(NARR12 = NARR12_Before + NARR12_During + NARR12_After) %>%
  # Set count variables with values of 0 to NA when that period was unobserved.
  mutate(across(.cols = c("NARR12_Before"),
                .fns = ~ifelse(Years_Before == 0, yes = NA, no = .))) %>%
  mutate(across(.cols = c("NARR12_During"),
                .fns = ~ifelse(Years_During == 0, yes = NA, no = .))) %>%
  var_labels(NARR12 = "No. of arrest offense records from 12 main crime categories") ->
  IDNEW
```

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

6.7 Merge Testing Window Variables Onto Prosecutor Charge Records

The next step is to merge variables from *INCEW* onto the charge data in *CHG* so that we can classify each prosecutor charge record according to whether the incident associated with it occurred before, during, or after the earliest testing window for the offender by using *IWhen*. Then we subset down to the *CHG* records that have non-missing *WDate* values plus *CCat12_Any* = 1, then select only the variables we need to keep before saving a new tibble called *CHGEW*.

```
INCEW %>%
  # Retain only variables we need to merge into CHG.
  select(IID, ESAKID, ESAK_IID, WDate, TWindow, IWhen) %>%
  # Merge INCEW variables onto CHG rows.
  right_join(x = ., y = CHG, by = "IID") %>%
  # Retain only CHG records for offenders with non-missing WDate values.
  filter(is.na(WDate) == FALSE) %>%
  # Retain only CHG records for the main 12 crime categories, dropping 999.
  filter(CCat12_Any == 1) %>%
  # Retain only variables we need, in sensible order.
  select(all_of(c(names(CHG), twvars))) %>%
  # Add a variable label that was otherwise being dropped.
  var_labels(TWindow = "Testing Window Date Interval") ->
  CHGEW
```

The original *CHG* tibble had 6052 prosecutor charge records. Dropping the records associated with offenders whose earliest testing window could not be identified or that were coded *CCat12_Any* = 0 because *CCat12* = 999 (excluded user-missing) leaves 5632 prosecutor charge records in *CHGEW*. Table 6 shows how many prosecutor charge records were associated with incidents that occurred before, during, and after the testing window.

```
# Table caption.
TCap <- paste("Frequency Distribution of When the Incident Associated With A",
              "Prosecutor Charge Record Occurred Relative to Testing Window")
# Footnote text.
FN <- paste("Only prosecutor charge records for incidents with valid testing",
            "window start dates and offenses from the 12 crime categories",
            "of interest were included. IWhen, when was incident relative",
            "to testing window?")

# Vector of text values for column labels to be used in the table.
clabels <- c("IWhen", "Frequency", "Percent")

xtabs(~IWhen, data = CHGEW) %>%
  addmargins() %>%
  as.data.frame() %>%
  mutate(Percent = 100*Freq/nrow(CHGEW)) %>%
  kable(., format = "latex", booktabs = TRUE, caption = TCap, digits = 2,
        col.names = clabels, format.args = list(big.mark = ",")) %>%
  column_spec(1:3, width = "1.5cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

Now, we need to get offender-level counts of prosecutor charge records associated with incidents that occurred before, during, and after the testing window respectively called *NCHG12_Before*, *NCHG12_During*, and *NCHG12_After*. We can aggregate as follows to get them. We merge these variables into *IDNEW* too. These new aggregated variables respect the criterion of counting only offenses in the 12 main crime categories of interest because of the case selection that went into building the *CHGEW* data.

```
CHGEW %>%
  mutate(NCHG12 = 1) %>%
  group_by(OID, IWhen, .drop = FALSE) %>%
  # Aggregate by OID and IWhen to get count variables
  summarize(NCHG12 = sum(NCHG12)) %>%
  pivot_wider(names_from = IWhen, values_from = NCHG12,
              names_glue = "NCHG12_{IWhen}") %>%
  var_labels(NCHG12_Before = "No. of prosecutor charge records from 12 main crime categories before testing window",
```

IWhen	Frequency	Percent
Before	1,488	26.42
During	407	7.23
After	3,737	66.35
Sum	5,632	100.00

Note:
Only prosecutor charge records for incidents with valid testing window start dates and offenses from the 12 crime categories of interest were included. IWhen, when was incident relative to testing window?

Table 6: Frequency Distribution of When the Incident Associated With A Prosecutor Charge Record Occurred Relative to Testing Window

```
            NCHG12_During = "No. of prosecutor charge records from 12 main crime categories during testing window",
            NCHG12_After  = "No. of prosecutor charge records from 12 main crime categories after testing window") %>%
  # Merge charge record count variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(c("NCHG12_Before", "NCHG12_During", "NCHG12_After")),
                .fns = replace_na, replace = 0)) %>%
  # Compute alternate NCHG for records from 12 main crime categories.
  mutate(NCHG12 = NCHG12_Before + NCHG12_During + NCHG12_After) %>%
  # Set count variables with values of 0 to NA when that period was unobserved.
  mutate(across(.cols = c("NCHG12_Before"),
                .fns = ~ifelse(Years_Before == 0, yes = NA, no = .))) %>%
  mutate(across(.cols = c("NCHG12_During"),
                .fns = ~ifelse(Years_During == 0, yes = NA, no = .))) %>%
  var_labels(NCHG12 = "No of prosecutor charge records from 12 main crime categories") ->
  IDNEW
```

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

6.8 Merge Testing Window Variables Onto Judicial Charge Records

The next step is to merge variables from *INCEW* onto the charge data in *JUD* so that we can classify each judicial charge record according to whether the incident associated with it occurred before, during, or after the earliest testing window for the offender by using *IWhen*. Then we subset down to the *JUD* records that have non-missing *WDate* values plus *JCat12_Any* = 1, then select only the variables we need to keep before saving a new tibble called *JUDEW*.

```
INCEW %>%
  # Retain only variables we need to merge into JUD.
  select(IID, ESAKID, ESAK_IID, WDate, TWindow, IWhen) %>%
  # Merge INCEW variables onto JUD rows.
  right_join(x = ., y = JUD, by = "IID") %>%
  # Retain only JUD records for offenders with non-missing WDate values.
  filter(is.na(WDate) == FALSE) %>%
  # Retain only JUD records for the main 12 crime categories, dropping 999.
  filter(JCat12_Any == 1) %>%
  # Retain only variables we need, in sensible order.
  select(all_of(c(names(JUD), twvars))) %>%
  # Add a variable label that was otherwise being dropped.
  var_labels(TWindow = "Testing Window Date Interval") ->
  JUDEW
```

The original *JUD* tibble had 12522 judicial charge records. Dropping the records associated with offenders whose earliest testing window could not be identified or that were coded *JCat12_Any* = 0 because *JCat12* = 999 (excluded user-missing) leaves 10995 judicial charge records in *JUDEW*. Table 7 shows how many judicial charge records were associated with incidents that occurred before, during, and after the testing window.

```
# Table caption.
TCap <- paste("Frequency Distribution of When the Incident Associated With A",
              "Judicial Charge Record Occurred Relative to Testing Window")
# Footnote text.
FN <- paste("Only judicial charge records for incidents with valid testing",
            "window start dates and offenses from the 12 crime categories",
            "of interest were included. IWhen, when was incident relative",
            "to testing window?")

# Vector of text values for column labels to be used in the table.
clabels <- c("IWhen", "Frequency", "Percent")

xtabs(~IWhen, data = JUDEW) %>%
  addmargins() %>%
  as.data.frame() %>%
  mutate(Percent = 100*Freq/nrow(JUDEW)) %>%
  kable(., format = "latex", booktabs = TRUE, caption = TCap, digits = 2,
        col.names = clabels, format.args = list(big.mark = ",")) %>%
  column_spec(1:3, width = "1.5cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

IWhen	Frequency	Percent
Before	3,098	28.18
During	1,065	9.69
After	6,832	62.14
Sum	10,995	100.00

Note:
Only judicial charge records for incidents with valid testing window start dates and offenses from the 12 crime categories of interest were included. IWhen, when was incident relative to testing window?

Table 7: Frequency Distribution of When the Incident Associated With A Judicial Charge Record Occurred Relative to Testing Window

Now, we need to get offender-level counts of judicial charge records associated with incidents that occurred before, during, and after the testing window respectively called *NJUD12_Before*, *NJUD12_During*, and *NJUD12_After*. We can aggregate as follows to get them. We merge these variables into *IDNEW* too. These new aggregated variables respect the criterion of counting only offenses in the 12 main crime categories of interest because of the case selection that went into building the *JUDEW* data.

```
JUDEW %>%
  mutate(NJUD12 = 1) %>%
  group_by(OID, IWhen, .drop = FALSE) %>%
  # Aggregate by OID and IWhen to get count variables
  summarize(NJUD12 = sum(NJUD12)) %>%
  pivot_wider(names_from = IWhen, values_from = NJUD12,
              names_glue = "NJUD12_{IWhen}") %>%
  var_labels(NJUD12_Before = "No. of adjudicated charge records from 12 main crime categories before testing window",
             NJUD12_During = "No. of adjudicated charge records from 12 main crime categories during testing window",
             NJUD12_After  = "No. of adjudicated charge records from 12 main crime categories after testing window") %>%
  # Merge adjudicated charge record count variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(c("NJUD12_Before", "NJUD12_During", "NJUD12_After")),
                .fns = replace_na, replace = 0)) %>%
  # Compute alternate NJUD for records from 12 main crime categories.
  mutate(NJUD12 = NJUD12_Before + NJUD12_During + NJUD12_After) %>%
  # Set count variables with values of 0 to NA when that period was unobserved.
  mutate(across(.cols = c("NJUD12_Before"),
                .fns = ~ifelse(Years_Before == 0, yes = NA, no = .))) %>%
```

```
  mutate(across(.cols = c("NJUD12_During"),
                .fns = ~ifelse(Years_During == 0, yes = NA, no = .))) %>%
  var_labels(NJUD12 = "No of judicial charge records from 12 main crime categories")->
  IDNEW
```

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

6.9 Merge Testing Window Variables Onto Conviction Records

The next step is to merge variables from *INCEW* onto the charge data in *CON* so that we can classify each conviction record according to whether the incident associated with it occurred before, during, or after the earliest testing window for the offender by using *IWhen*. Then we subset down to the *CON* records that have non-missing *WDate* values plus *JCat12_Any* = 1, then select only the variables we need to keep before saving a new tibble called *CONEW*.

```
INCEW %>%
  # Retain only variables we need to merge into CON.
  select(IID, ESAKID, ESAK_IID, WDate, TWindow, IWhen) %>%
  # Merge INCEW variables onto CON rows.
  right_join(x = ., y = CON, by = "IID") %>%
  # Retain only CON records for offenders with non-missing WDate values.
  filter(is.na(WDate) == FALSE) %>%
  # Retain only CON records for the main 12 crime categories, dropping 999.
  filter(JCat12_Any == 1) %>%
  # Retain only variables we need, in sensible order.
  select(all_of(c(names(JUD), twvars))) %>%
  # Add a variable label that was otherwise being dropped.
  var_labels(TWindow = "Testing Window Date Interval") ->
  CONEW
```

The original *CON* tibble had 6971 conviction records. Dropping the records associated with offenders whose earliest testing window could not be identified or that were coded *JCat12_Any* = 0 because *JCat12* = 999 (excluded user-missing) leaves 6021 judicial charge records in *CONEW*. Table 8 shows how many conviction records were associated with incidents that occurred before, during, and after the testing window.

```
# Table caption.
TCap <- paste("Frequency Distribution of When the Incident Associated With A",
              "Conviction Record Occurred Relative to Testing Window")
# Footnote text.
FN <- paste("Only conviction records for incidents with valid testing",
            "window start dates and offenses from the 12 crime categories",
            "of interest were included. IWhen, when was incident relative",
            "to testing window?")

# Vector of text values for column labels to be used in the table.
clabels <- c("IWhen", "Frequency", "Percent")

xtabs(~IWhen, data = CONEW) %>%
  addmargins() %>%
  as.data.frame() %>%
  mutate(Percent = 100*Freq/nrow(CONEW)) %>%
  kable(., format = "latex", booktabs = TRUE, caption = TCap, digits = 2,
        col.names = clabels, format.args = list(big.mark = ",")) %>%
  column_spec(1:3, width = "1.5cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
         threeparttable = TRUE)
```

Now, we need to get offender-level counts of conviction records associated with incidents that occurred before, during, and after the testing window respectively called *NCON12_Before*, *NCON12_During*, and *NCON12_After*. We can aggregate as follows to get them. We merge these variables into *IDNEW* too. These new aggregated variables respect the criterion of counting only offenses in the 12 main crime categories of interest because of the case selection that went into building the *CONEW* data.

IWhen	Frequency	Percent
Before	1,695	28.15
During	515	8.55
After	3,811	63.30
Sum	6,021	100.00

Note: Only conviction records for incidents with valid testing window start dates and offenses from the 12 crime categories of interest were included. IWhen, when was incident relative to testing window?

Table 8: Frequency Distribution of When the Incident Associated With A Conviction Record Occurred Relative to Testing Window

```
CONEW %>%
  mutate(NCON12 = 1) %>%
  group_by(OID, IWhen, .drop = FALSE) %>%
  # Aggregate by OID and IWhen to get count variables
  summarize(NCON12 = sum(NCON12)) %>%
  pivot_wider(names_from = IWhen, values_from = NCON12,
              names_glue = "NCON12_{IWhen}") %>%
  var_labels(NCON12_Before = "No. of conviction records from 12 main crime categories before testing window",
             NCON12_During = "No. of conviction records from 12 main crime categories during testing window",
             NCON12_After  = "No. of conviction records from 12 main crime categories after testing window") %>%
  # Merge convicted charge record count variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(c("NCON12_Before", "NCON12_During", "NCON12_After")),
                .fns = replace_na, replace = 0)) %>%
  # Compute alternate NCON for records from 12 main crime categories.
  mutate(NCON12 = NCON12_Before + NCON12_During + NCON12_After) %>%
  # Set count variables with values of 0 to NA when that period was unobserved.
  mutate(across(.cols = c("NCON12_Before"),
                .fns = ~ifelse(Years_Before == 0, yes = NA, no = .))) %>%
  mutate(across(.cols = c("NCON12_During"),
                .fns = ~ifelse(Years_During == 0, yes = NA, no = .))) %>%
  var_labels(NCON12 = "No of conviction records from 12 main crime categories")->
  IDNEW
```

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

6.10 Aggregating Crime Category Dummy Variables to Incident Level Dummy Variables

We used the *dcode()* function to create the dummy coded crime category variables while importing data from the SPSS data files into the *ARR*, *CHG*, and *CON* tibbles, which were each subsequently modified and saved as the *ARREW*, *CHGEW*, and *CONEW* tibbles.

In the *ARREW* tibble, the variables *ACat12_1* to *ACat12_12* and *ACat12_999* mark which category of crime the offender was arrested for on a given arrest offense record. The corresponding *CCat12_1* to *CCat12_12* and *CCat12_999* variables in *CHGEW* show which crime category the individual was charged with by a prosecuting attorney on a charge record. Meanwhile, *JCat12_1* to *JCat12_12* and *JCat12_999* variables in *CONEW* show which crime category the offender was convicted of on a given conviction record.

Similarly, the *ACat12_Any*, *CCat12_Any*, and *JCat12_Any* dummy code variables mark *ARREW*, *CHGEW*, and *CONEW* records that show an offender was respectively arrested for, charged with, or convicted of one of the 12 main crime categories of interest.

All three of those tibbles can have multiple records per criminal incident. This section aggregates those dummy coded crime category variables from the record level to the incident level to flag presence or absence of each crime category on the incident.

It merges the resulting variables onto the *INCEW* tibble. That allows us to see which incidents were associated with arrests, charges, and/or convictions for each category of crime or for any of the 12 main categories.

The code chunk below creates objects containing vectors of variable names that can be used to simplify subsequent code.

```
# Create objects with vectors of variable names we need later.
avars      <- paste("ACat12", c(1:12, 999, "Any", "Sexual", "Violent",
                                "Property", "Other"), sep = "_")
avarsb     <- paste0(avars, "_Before")
avarsd     <- paste0(avars, "_During")
avarsa     <- paste0(avars, "_After")
avarsbda   <- c(avarsb, avarsd, avarsa)
cvars      <- paste("CCat12", c(1:12, 999, "Any", "Sexual", "Violent",
                                "Property", "Other"), sep = "_")
cvarsb     <- paste0(cvars, "_Before")
cvarsd     <- paste0(cvars, "_During")
cvarsa     <- paste0(cvars, "_After")
cvarsbda   <- c(cvarsb, cvarsd, cvarsa)
jvars      <- paste("JCat12", c(1:12, 999, "Any", "Sexual", "Violent",
                                "Property", "Other"), sep = "_")
jvarsb     <- paste0(jvars, "_Before")
jvarsd     <- paste0(jvars, "_During")
jvarsa     <- paste0(jvars, "_After")
jvarsbda   <- c(jvarsb, jvarsd, jvarsa)
hxvars     <- paste("HXCat12", c(1:12, 999, "Any", "Sexual", "Violent",
                                "Property", "Other"), sep = "_")
hxvarsb    <- paste0(hxvars, "_Before")
hxvarsd    <- paste0(hxvars, "_During")
hxvarsa    <- paste0(hxvars, "_After")
hxvarsbda  <- c(hxvarsb, hxvarsd, hxvarsa)
```

As a reminder, the crime categories are shown in Table 9, along with corresponding variable names.

```
# Table caption.
TCap <- paste("Crime Category Labels and Names for Dummy Coded Variables From",
              "ARREW, CHGEW, and CONEW Data To be Merged Into INCEW Data")
# Footnote text.
FN <- paste("ARREW, variables pertaining to arrests;",
            "CHGEW, variables pertaining to prosecutor charges;",
            "CONEW, variables pertaining to convictions.")

# Create data for table.
CrimeCats <- data.frame(Label = c(get_labels(ARREW$ACat12),
                                  "Any of 12 main categories (arson to weapons)",
                                  "Any sexual crimes (2 categories)",
                                  "Any violent crimes (5 categories)",
                                  "Any property crimes (3 categories)",
                                  "Any other crimes (2 categories)"),
                        ARREW = avars,
                        CHGEW = cvars,
                        CONEW = jvars)

kable(CrimeCats, format = "latex", booktabs = TRUE, caption = TCap) %>%
  add_header_above(c(" " = 1, "Source Datasets and Variable Names" = 3)) %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

6.10.1 Incidents With Arrests

For any given incident (identified by a unique value of *IID*), there can be multiple arrest offense records in *ARREW*. The dummy variables *ACat12_1* to *ACat12_12* and *ACat12_999* each mark whether or not a particular arrest offense record was associated with a specific type of crime, while *ACat12_Any* marks whether or not the arrest offense was associated any of the 12 main categories of crime. Here, we aggregate them up to the incident level by taking the maximum observed value for each incident. The resulting incident-level records will have values of 1 for every crime category for which the offender was arrested due to that incident and values of 0 for every crime category for which the offender was not arrested due to that incident.

	Source Datasets and Variable Names		
Label	ARREW	CHGEW	CONEW
Arson	ACat12_1	CCat12_1	JCat12_1
Assault - DV, stalking	ACat12_2	CCat12_2	JCat12_2
Assault - non-sexual, non-DV	ACat12_3	CCat12_3	JCat12_3
Burglary	ACat12_4	CCat12_4	JCat12_4
Criminal sexual conduct	ACat12_5	CCat12_5	JCat12_5
Drug crime	ACat12_6	CCat12_6	JCat12_6
Homicide	ACat12_7	CCat12_7	JCat12_7
Larceny/Theft/Fraud	ACat12_8	CCat12_8	JCat12_8
Robbery	ACat12_9	CCat12_9	JCat12_9
Sex crime, other (excluding CSC)	ACat12_10	CCat12_10	JCat12_10
Traffic & Ordinances	ACat12_11	CCat12_11	JCat12_11
Weapons	ACat12_12	CCat12_12	JCat12_12
Excluded user-missing	ACat12_999	CCat12_999	JCat12_999
Any of 12 main categories (arson to weapons)	ACat12_Any	CCat12_Any	JCat12_Any
Any sexual crimes (2 categories)	ACat12_Sexual	CCat12_Sexual	JCat12_Sexual
Any violent crimes (5 categories)	ACat12_Violent	CCat12_Violent	JCat12_Violent
Any property crimes (3 categories)	ACat12_Property	CCat12_Property	JCat12_Property
Any other crimes (2 categories)	ACat12_Other	CCat12_Other	JCat12_Other

Note: ARREW, variables pertaining to arrests; CHGEW, variables pertaining to prosecutor charges; CONEW, variables pertaining to convictions.

Table 9: Crime Category Labels and Names for Dummy Coded Variables From ARREW, CHGEW, and CONEW Data To be Merged Into INCEW Data

```
# Aggregate ACat12 dummy codes by IID to incident level binary variables.
ARREW %>%
  group_by(IID) %>%
  summarize(across(.cols = all_of(avars), .fns = max)) %>%
  # Merge dummy variables onto INCEW.
  left_join(x = INCEW, y = ., by = "IID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(avars), .fns = replace_na, replace = 0)) %>%
  var_labels(ACat12_1   = "Arrested for arson",
             ACat12_2   = "Arrested for assault - DV, stalking",
             ACat12_3   = "Arrested for assault - non-sexual, non-DV",
             ACat12_4   = "Arrested for burglary",
             ACat12_5   = "Arrested for criminal sexual conduct",
             ACat12_6   = "Arrested for drug crime",
             ACat12_7   = "Arrested for homicide",
             ACat12_8   = "Arrested for larceny/theft/fraud",
             ACat12_9   = "Arrested for robbery",
             ACat12_10  = "Arrested for sex crime, other excluding CSC",
             ACat12_11  = "Arrested for traffic/ordinances",
             ACat12_12  = "Arrested for weapons",
             ACat12_999 = "Arrested for excluded user-missing",
             ACat12_Any = "Arrested for any of 12 main crime categories",
             ACat12_Sexual = "Arrested for any sexual crimes (2 categories)",
             ACat12_Violent = "Arrested for any violent crimes (5 categories)",
             ACat12_Property = "Arrested for any property crimes (3 categories)",
             ACat12_Other = "Arrested for any other crimes (2 categories)") ->
  INCEW
```

6.10.2 Incidents With Charges

For any given incident (identified by a unique value of *IID*), there can be multiple prosecutor charges records in *CHGEW*. The dummy variables *CCat12_1* to *CCat12_12* and *CCat12_999* each mark whether or not a particular prosecutor charge

record was associated with a specific type of crime, while *CCat12_Any* marks whether or not the charge was associated any of the 12 main categories of crime. Here, we aggregate them up to the incident level by taking the maximum observed value for each incident. The resulting incident-level records will have values of 1 for every crime category for which the offender was charged due to that incident and values of 0 for every crime category for which the offender was not charged due to that incident.

```
# Aggregate CCat12 dummy codes by IID to incident level binary variables.
CHGEW %>%
  group_by(IID) %>%
  summarize(across(.cols = all_of(cvars), .fns = max))  %>%
  # Merge dummy variables onto INCEW.
  left_join(x = INCEW, y = ., by = "IID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(cvars), .fns = replace_na, replace = 0)) %>%
  var_labels(CCat12_1   = "Charged for arson",
             CCat12_2   = "Charged for assault - DV, stalking",
             CCat12_3   = "Charged for assault - non-sexual, non-DV",
             CCat12_4   = "Charged for burglary",
             CCat12_5   = "Charged for criminal sexual conduct",
             CCat12_6   = "Charged for drug crime",
             CCat12_7   = "Charged for homicide",
             CCat12_8   = "Charged for larceny/theft/fraud",
             CCat12_9   = "Charged for robbery",
             CCat12_10  = "Charged for sex crime, other excluding CSC",
             CCat12_11  = "Charged for traffic/ordinances",
             CCat12_12  = "Charged for weapons",
             CCat12_999 = "Charged for excluded user-missing",
             CCat12_Any = "Charged for any of 12 main crime categories",
             CCat12_Sexual = "Charged for any sexual crimes (2 categories)",
             CCat12_Violent = "Charged for any violent crimes (5 categories)",
             CCat12_Property = "Charged for any property crimes (3 categories)",
             CCat12_Other = "Charged for any other crimes (2 categories)") ->
  INCEW
```

6.10.3 Incidents With Convictions

For any given incident (identified by a unique value of *IID*), there can be multiple conviction records in *CONEW*. The dummy variables *JCat12_1* to *JCat12_12* and *JCat12_999* each mark whether or not a particular adjudicated conviction charge record was associated with a specific type of crime, while *JCat12_Any* marks whether or not the charge was associated any of the 12 main categories of crime. Here, we aggregate them up to the incident level by taking the maximum observed value for each incident. The resulting incident-level records will have values of 1 for every crime category for which the offender was convicted due to that incident and values of 0 for every crime category for which the offender was not convicted due to that incident.

```
# Aggregate JCat12 dummy codes by IID to incident level binary variables.
CONEW %>%
  group_by(IID) %>%
  summarize(across(.cols = all_of(jvars), .fns = max)) %>%
  # Merge dummy variables onto INCEW.
  left_join(x = INCEW, y = ., by = "IID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(jvars), .fns = replace_na, replace = 0)) %>%
  var_labels(JCat12_1   = "Convicted for arson",
             JCat12_2   = "Convicted for assault - DV, stalking",
             JCat12_3   = "Convicted for assault - non-sexual, non-DV",
             JCat12_4   = "Convicted for burglary",
             JCat12_5   = "Convicted for criminal sexual conduct",
             JCat12_6   = "Convicted for drug crime",
             JCat12_7   = "Convicted for homicide",
             JCat12_8   = "Convicted for larceny/theft/fraud",
             JCat12_9   = "Convicted for robbery",
             JCat12_10  = "Convicted for sex crime, other excluding CSC",
             JCat12_11  = "Convicted for traffic/ordinances",
             JCat12_12  = "Convicted for weapons",
             JCat12_999 = "Convicted for excluded user-missing",
             JCat12_Any = "Convicted for any of 12 main crime categories",
             JCat12_Sexual = "Convicted for any sexual crimes (2 categories)",
```

```
            JCat12_Violent = "Convicted for any violent crimes (5 categories)",
            JCat12_Property = "Convicted for any property crimes (3 categories)",
            JCat12_Other = "Convicted for any other crimes (2 categories)") ->
  INCEW
```

6.11 Create Incident-Level Crime Category History Variables

For each crime category, the *INCEW* dataset now contains three separate dummy variables showing whether the offender was (a) arrested, (b) charged, or (c) convicted of the offense associated with that category as a result of that incident. However, we still need to create a single dummy variable for each crime category that combines data across those three types of criminal history events. The resulting set of history variables will be called *HXCat12_1* to *HXCat12_12* and *HXCat12_999*. Values of 1 will mean the CHR data shows that the offender was *arrested for, charged with, or convicted of* the offense in question for that incident. It does not matter which of the three events occurred: any one of them (and any combination of them) is sufficient to set this value to 1. In contrast, a value of 0 will mean the CHR data shows that the offender was *not arrested for, not charged with, and not convicted of* the offense in question for that incident. The new variables will be added to the *INCEW* dataset.

Meanwhile, the *HXCat12_Any* variable will show whether the offender was *arrested for, charged with, or convicted of* any of the 12 main categories of crime.

```
INCEW %>%
  # Create history variables
  mutate(HXCat12_1 = if_else(ACat12_1 + CCat12_1 + JCat12_1 >= 1,
                             true = 1, false = 0),
         HXCat12_2 = if_else(ACat12_2 + CCat12_2 + JCat12_2 >= 1,
                             true = 1, false = 0),
         HXCat12_3 = if_else(ACat12_3 + CCat12_3 + JCat12_3 >= 1,
                             true = 1, false = 0),
         HXCat12_4 = if_else(ACat12_4 + CCat12_4 + JCat12_4 >= 1,
                             true = 1, false = 0),
         HXCat12_5 = if_else(ACat12_5 + CCat12_5 + JCat12_5 >= 1,
                             true = 1, false = 0),
         HXCat12_6 = if_else(ACat12_6 + CCat12_6 + JCat12_6 >= 1,
                             true = 1, false = 0),
         HXCat12_7 = if_else(ACat12_7 + CCat12_7 + JCat12_7 >= 1,
                             true = 1, false = 0),
         HXCat12_8 = if_else(ACat12_8 + CCat12_8 + JCat12_8 >= 1,
                             true = 1, false = 0),
         HXCat12_9 = if_else(ACat12_9 + CCat12_9 + JCat12_9 >= 1,
                             true = 1, false = 0),
         HXCat12_10 = if_else(ACat12_10 + CCat12_10 + JCat12_10 >= 1,
                             true = 1, false = 0),
         HXCat12_11 = if_else(ACat12_11 + CCat12_11 + JCat12_11 >= 1,
                             true = 1, false = 0),
         HXCat12_12 = if_else(ACat12_12 + CCat12_12 + JCat12_12 >= 1,
                             true = 1, false = 0),
         HXCat12_999 = if_else(ACat12_999 + CCat12_999 + JCat12_999 >= 1,
                             true = 1, false = 0),
         HXCat12_Any = if_else(ACat12_Any + CCat12_Any + JCat12_Any >= 1,
                             true = 1, false = 0),
         HXCat12_Sexual = if_else(ACat12_Sexual + CCat12_Sexual + JCat12_Sexual >= 1,
                                  true = 1, false = 0),
         HXCat12_Violent = if_else(ACat12_Violent + CCat12_Violent + JCat12_Violent >= 1,
                                   true = 1, false = 0),
         HXCat12_Property = if_else(ACat12_Property + CCat12_Property + JCat12_Property >= 1,
                                    true = 1, false = 0),
         HXCat12_Other = if_else(ACat12_Other + CCat12_Other + JCat12_Other >= 1,
                                 true = 1, false = 0)) %>%
  var_labels(HXCat12_1 = "History shows arrested, charged, or convicted for arson",
             HXCat12_2 = "History shows arrested, charged, or convicted for assault - DV, stalking arrest",
             HXCat12_3 = "History shows arrested, charged, or convicted for assault - non-sexual, non-DV",
             HXCat12_4 = "History shows arrested, charged, or convicted for burglary",
             HXCat12_5 = "History shows arrested, charged, or convicted for criminal sexual conduct",
             HXCat12_6 = "History shows arrested, charged, or convicted for drug crime",
             HXCat12_7 = "History shows arrested, charged, or convicted for homicide",
             HXCat12_8 = "History shows arrested, charged, or convicted for larceny/theft/fraud ",
             HXCat12_9 = "History shows arrested, charged, or convicted for robbery",
```

```
            HXCat12_10 = "History shows arrested, charged, or convicted for sex crime - other, excludinng CSC",
            HXCat12_11 = "History shows arrested, charged, or convicted for traffic/ordinances",
            HXCat12_12 = "History shows arrested, charged, or convicted for weapons",
            HXCat12_999 = "History shows arrested, charged, or convicted for excluded user-missing",
            HXCat12_Any = "History shows arrested, charged, or convicted for any of 12 main crime categories",
            HXCat12_Sexual = "History shows arrested, charged, or convicted for any sexual crimes (2 categories)",
            HXCat12_Violent = "History shows arrested, charged, or convicted for any violent crimes (5 categories)",
            HXCat12_Property = "History shows arrested, charged, or convicted for any property crimes (3 categories)",
            HXCat12_Other = "History shows arrested, charged, or convicted for any other crimes (2 categories)") ->
  INCEW
```

Note that *HXCat12_5* is related to the existing variable *CSC_ANY* variable, but a slightly more stringent definition because *HXCat12_5* codes incidents that had adjudicated charges for CSC that were not convictions (and had no CSC arrests and no CSC prosecutor charges) as 0, whereas *CSC_ANY* would code them as 1. We will use *HXCat12_5* in the analyses for this paper instead of *CSC_ANY*. Table 10 shows how these two variables cross-tabulate.

```
# Table caption.
TCap <- paste0("Crosstabulate Incident-Level Criminal Sexual Conduct History ",
               "Variables (N = ", nrow(INCEW), ")")
# Footnote text.
FN <- paste("Only incident records for incidents with valid testing window",
            "start dates were included. The variables disagree when the only",
            "CHR data for CSC associated with the incident are adjudicated",
            "charge records that are not convictions.")

# Vector of text values for column labels to be used in the table.
clabels <- c("HXCat12_5", "CSC_ANY", "Frequency", "Percent")

xtabs(~HXCat12_5 + CSC_ANY, data = INCEW) %>%
  as.data.frame() %>%
  mutate(Percent = 100*Freq/nrow(INCEW)) %>%
  kable(., format = "latex", booktabs = TRUE, caption = TCap, digits = 2,
        col.names = clabels, format.args = list(big.mark = ",")) %>%
  column_spec(1:3, width = "1.5 cm") %>%
  footnote(general = FN, general_title = "Note: ", footnote_as_chunk = TRUE,
           threeparttable = TRUE)
```

HXCat12_5	CSC_ANY	Frequency	Percent
0	0	8,372	92.23
1	0	0	0.00
0	1	28	0.31
1	1	677	7.46

Note: Only incident records for incidents with valid testing window start dates were included. The variables disagree when the only CHR data for CSC associated with the incident are adjudicated charge records that are not convictions.

Table 10: Crosstabulate Incident-Level Criminal Sexual Conduct History Variables (N = 9077)

6.12 Filter Incident Records

Now we need to subset the *INCEW* data to contain only records for which *HXCat12_Any* = 1 so that we only retain incidents where the offender was *arrested for, charged with, or convicted of* one or more of the 12 main crime categories.

```
INCEW %>%
  filter(HXCat12_Any == 1) ->
  INCEW
```

That reduces *INCEW* to 8588 records.

6.13 Aggregating Crime Category Dummy Variables to Offender Level Count Variables

Here we aggregate the dummy coded crime category variables data from the incident level to offender level incident counts for each crime category.

6.13.1 Incidents With Arrests (Overall Plus Before, During, and After Testing Window)

For any given offender (identifed by an *OID* value), there can be multiple incidents (identified by *IID* values). Aggregating the dummy variables ACat12_1 to ACat12_12 and ACat12_999 by summing across incidents for each offender allows us to get overall incident counts showing how often the offender was arrested for various types of crime over the course of the entire (documented) criminal history. The aggregated variables merged into *IDNEW* retain the same variable names as the source dummy-coded variables.

```
# Aggregate ACat12 dummy codes to offender level incident counts.
INCEW %>%
  group_by(OID) %>%
  # Aggregate dummy codes by OID into incident count variables
  summarize(across(.cols = all_of(avars), .fns = sum))  %>%
  # Merge dummy variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(avars), .fns = replace_na, replace = 0)) %>%
  var_labels(ACat12_1 = "No. incidents arrested for arson",
             ACat12_2 = "No. incidents arrested for assault - DV, stalking",
             ACat12_3 = "No. incidents arrested for assault - non-sexual, non-DV",
             ACat12_4 = "No. incidents arrested for burglary",
             ACat12_5 = "No. incidents arrested for criminal sexual conduct",
             ACat12_6 = "No. incidents arrested for drug crime",
             ACat12_7 = "No. incidents arrested for homicide",
             ACat12_8 = "No. incidents arrested for larceny/theft/fraud",
             ACat12_9 = "No. incidents arrested for robbery",
             ACat12_10 = "No. incidents arrested for sex crime - other, excluding CSC",
             ACat12_11 = "No. incidents arrested for traffic/ordinances",
             ACat12_12 = "No. incidents arrested for weapons",
             ACat12_999 = "No. incidents arrested for excluded user-missing",
             ACat12_Any = "No. incidents arrested for any of 12 main crime categories",
             ACat12_Sexual = "No. incidents arrested for any sexual crimes (2 categories)",
             ACat12_Violent = "No. incidents arrested for any violent crimes (5 categories)",
             ACat12_Property = "No. incidents arrested for any property crimes (3 categories)",
             ACat12_Other = "No. incidents arrested for any other crimes (2 categories)") ->
  IDNEW
```

We also want to get separate counts for each crime category broken down by when the incident occurred relative to the earliest testing window (*IWhen*). This next chunk does that. The variable naming convention just adds suffixes to the previously defined variable names to indicate the period to which each count variable pertains. Because some offenders' adult CHRs did not overlap with the before and during periods, we have to selectively recode counts of zero to missing data when the observed duration of the period in question was zero years. Otherwise we risk mistaking what amounts to a structural zero (no opportunity to observe incidents because because we had zero years of that period represented in the adult CHR) with a sampling zero (we had the opportunity to observe incidents because we observed more than 0 years of the period in the adult CHR, but just did not see any in that period).

```
# Aggregate ACat12 dummy codes to offender level incident counts by IWhen .
INCEW %>%
  group_by(OID, IWhen, .drop = FALSE) %>%
  pivot_wider(names_from = IWhen, values_from = all_of(avars),
              values_fill = 0) %>%
  group_by(OID) %>%
  # Aggregate dummy codes by OID into incident count variables
  summarize(across(.cols = all_of(avarsbda), .fns = sum, na.rm = TRUE))  %>%
  # Merge count variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Set count variables with values of 0 to NA when that period was unobserved.
  mutate(across(.cols = all_of(avarsb),
                .fns = ~ifelse(Years_Before == 0, yes = NA, no = .))) %>%
  mutate(across(.cols = all_of(avarsd),
```

```
              .fns = ~ifelse(Years_During == 0, yes = NA, no = .))) %>%
  var_labels(ACat12_1_Before = "No. incidents arrested for arson before testing window before testing window",
             ACat12_2_Before = "No. incidents arrested for assault - DV, stalking before testing window",
             ACat12_3_Before = "No. incidents arrested for assault - non-sexual, non-DV before testing window",
             ACat12_4_Before = "No. incidents arrested for burglary before testing window",
             ACat12_5_Before = "No. incidents arrested for criminal sexual conduct before testing window",
             ACat12_6_Before = "No. incidents arrested for drug crime before testing window",
             ACat12_7_Before = "No. incidents arrested for homicide before testing window",
             ACat12_8_Before = "No. incidents arrested for larceny/theft/fraud before testing window",
             ACat12_9_Before = "No. incidents arrested for robbery before testing window",
             ACat12_10_Before = "No. incidents arrested for sex crime - other, excluding CSC before testing window",
             ACat12_11_Before = "No. incidents arrested for traffic/ordinances before testing window",
             ACat12_12_Before = "No. incidents arrested for weapons before testing window",
             ACat12_999_Before = "No. incidents arrested for excluded user-missing before testing window",
             ACat12_Any_Before = "No. incidents arrested for any of 12 main crime categories before testing window",
             ACat12_Sexual_Before = "No. incidents arrested for any sexual crimes (2 categories) before testing window",
             ACat12_Violent_Before = "No. incidents arrested for any violent crimes (5 categories) before testing window",
             ACat12_Property_Before = "No. incidents arrested for any property crimes (3 categories) before testing window",
             ACat12_Other_Before = "No. incidents arrested for any other crimes (2 categories) before testing window",
             ACat12_1_During = "No. incidents arrested for arson during testing window",
             ACat12_2_During = "No. incidents arrested for assault - DV, stalking during testing window",
             ACat12_3_During = "No. incidents arrested for assault - non-sexual, non-DV during testing window",
             ACat12_4_During = "No. incidents arrested for burglary during testing window",
             ACat12_5_During = "No. incidents arrested for criminal sexual conduct during testing window",
             ACat12_6_During = "No. incidents arrested for drug crime during testing window",
             ACat12_7_During = "No. incidents arrested for homicide during testing window",
             ACat12_8_During = "No. incidents arrested for larceny/theft/fraud during testing window",
             ACat12_9_During = "No. incidents arrested for robbery during testing window",
             ACat12_10_During = "No. incidents arrested for sex crime - other, excluding CSC during testing window",
             ACat12_11_During = "No. incidents arrested for traffic/ordinances during testing window",
             ACat12_12_During = "No. incidents arrested for weapons during testing window",
             ACat12_999_During = "No. incidents arrested for excluded user-missing during testing window",
             ACat12_Any_During = "No. incidents arrested for any of 12 main crime categories during testing window",
             ACat12_Sexual_During = "No. incidents arrested for any sexual crimes (2 categories) during testing window",
             ACat12_Violent_During = "No. incidents arrested for any violent crimes (5 categories) during testing window",
             ACat12_Property_During = "No. incidents arrested for any property crimes (3 categories) during testing window",
             ACat12_Other_During = "No. incidents arrested for any other crimes (2 categories) during testing window",
             ACat12_1_After = "No. incidents arrested for arson after testing window",
             ACat12_2_After = "No. incidents arrested for assault - DV, stalking after testing window",
             ACat12_3_After = "No. incidents arrested for assault - non-sexual, non-DV after testing window",
             ACat12_4_After = "No. incidents arrested for burglary after testing window",
             ACat12_5_After = "No. incidents arrested for criminal sexual conduct after testing window",
             ACat12_6_After = "No. incidents arrested for drug crime after testing window",
             ACat12_7_After = "No. incidents arrested for homicide after testing window",
             ACat12_8_After = "No. incidents arrested for larceny/theft/fraud after testing window",
             ACat12_9_After = "No. incidents arrested for robbery after testing window",
             ACat12_10_After = "No. incidents arrested for sex crime - other, excluding CSC after testing window",
             ACat12_11_After = "No. incidents arrested for traffic/ordinances after testing window",
             ACat12_12_After = "No. incidents arrested for weapons after testing window",
             ACat12_999_After = "No. incidents arrested for excluded user-missing",
             ACat12_Any_After = "No. incidents arrested for any of 12 main crime categories after testing window",
             ACat12_Sexual_After = "No. incidents arrested for any sexual crimes (2 categories) after testing window",
             ACat12_Violent_After = "No. incidents arrested for any violent crimes (5 categories) after testing window",
             ACat12_Property_After = "No. incidents arrested for any property crimes (3 categories) after testing window",
             ACat12_Other_After = "No. incidents arrested for any other crimes (2 categories) after testing window") ->
  IDNEW
```

6.13.2 Incidents With Charges (Overall Plus Before, During, and After Testing Window)

For any given offender (identifed by an *OID* value), there can be multiple incidents (identified by *IID* values). Aggregating the dummy variables CCat12_1 to CCat12_12 and CCat12_999 by summing across incidents for each offender allows us to get overall incident counts showing how often the offender was charged for various types of crime over the course of the entire (documented) criminal history. The aggregated variables merged into *IDNEW* retain the same varable names as the source dummy-coded variables.

```
# Aggregate CCat12 dummy codes to offender level incident counts.
INCEW %>%
  group_by(OID) %>%
```

```
  # Aggregate dummy codes by OID into incident count variables
  summarize(across(.cols = all_of(cvars), .fns = sum))  %>%
  # Merge dummy variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(cvars), .fns = replace_na, replace = 0)) %>%
  var_labels(CCat12_1 = "No. incidents charged for arson",
             CCat12_2 = "No. incidents charged for assault - DV, stalking",
             CCat12_3 = "No. incidents charged for assault - non-sexual, non-DV",
             CCat12_4 = "No. incidents charged for burglary",
             CCat12_5 = "No. incidents charged for criminal sexual conduct",
             CCat12_6 = "No. incidents charged for drug crime",
             CCat12_7 = "No. incidents charged for homicide",
             CCat12_8 = "No. incidents charged for larceny/theft/fraud",
             CCat12_9 = "No. incidents charged for robbery",
             CCat12_10 = "No. incidents charged for sex crime - other, excluding CSC",
             CCat12_11 = "No. incidents charged for traffic/ordinances",
             CCat12_12 = "No. incidents charged for weapons",
             CCat12_999 = "No. incidents charged for excluded user-missing",
             CCat12_Any = "No. incidents charged for any of 12 main crime categories",
             CCat12_Sexual = "No. incidents charged for any sexual crimes (2 categories)",
             CCat12_Violent = "No. incidents charged for any violent crimes (5 categories)",
             CCat12_Property = "No. incidents charged for any property crimes (3 categories)",
             CCat12_Other = "No. incidents charged for any other crimes (2 categories)") ->
  IDNEW
```

We also want to get separate counts for each crime category broken down by when the incident occurred relative to the earliest testing window (*IWhen*). This next chunk does that. The variable naming convention just adds suffixes to the previously defined variable names to indicate the period to which each count variable pertains. Because some offenders' adult CHRs did not overlap with the before and during periods, we have to selectively recode counts of zero to missing data when the observed duration of the period in question was zero years. Otherwise we risk mistaking what amounts to a structural zero (no opportunity to observe incidents because because we had zero years of that period represented in the adult CHR) with a sampling zero (we had the opportunity to observe incidents because we observed more than 0 years of the period in the adult CHR, but just did not see any in that period).

```
# Aggregate CCat12 dummy codes to offender level incident counts by IWhen .
INCEW %>%
  group_by(OID, IWhen, .drop = FALSE) %>%
  pivot_wider(names_from = IWhen, values_from = all_of(cvars),
              values_fill = 0) %>%
  group_by(OID) %>%
  # Aggregate dummy codes by OID into incident count variables
  summarize(across(.cols = all_of(cvarsbda), .fns = sum, na.rm = TRUE))  %>%
  # Merge count variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Set count variables with values of 0 to NA when that period was unobserved.
  mutate(across(.cols = all_of(cvarsb),
                .fns = ~ifelse(Years_Before == 0, yes = NA, no = .))) %>%
  mutate(across(.cols = all_of(cvarsd),
                .fns = ~ifelse(Years_During == 0, yes = NA, no = .))) %>%
  var_labels(CCat12_1_Before = "No. incidents charged for arson before testing window before testing window",
             CCat12_2_Before = "No. incidents charged for assault - DV, stalking before testing window",
             CCat12_3_Before = "No. incidents charged for assault - non-sexual, non-DV before testing window",
             CCat12_4_Before = "No. incidents charged for burglary before testing window",
             CCat12_5_Before = "No. incidents charged for criminal sexual conduct before testing window",
             CCat12_6_Before = "No. incidents charged for drug crime before testing window",
             CCat12_7_Before = "No. incidents charged for homicide before testing window",
             CCat12_8_Before = "No. incidents charged for larceny/theft/fraud before testing window",
             CCat12_9_Before = "No. incidents charged for robbery before testing window",
             CCat12_10_Before = "No. incidents charged for sex crime - other, excluding CSC before testing window",
             CCat12_11_Before = "No. incidents charged for traffic/ordinances before testing window",
             CCat12_12_Before = "No. incidents charged for weapons before testing window",
             CCat12_999_Before = "No. incidents charged for excluded user-missing before testing window",
             CCat12_Any_Before = "No. incidents charged for any of 12 main crime categories before testing window",
             CCat12_Sexual_Before = "No. incidents charged for any sexual crimes (2 categories) before testing window",
             CCat12_Violent_Before = "No. incidents charged for any violent crimes (5 categories) before testing window",
             CCat12_Property_Before = "No. incidents charged for any property crimes (3 categories) before testing window",
             CCat12_Other_Before = "No. incidents charged for any other crimes (2 categories) before testing window",
             CCat12_1_During = "No. incidents charged for arson during testing window",
```

```
            CCat12_2_During = "No. incidents charged for assault - DV, stalking during testing window",
            CCat12_3_During = "No. incidents charged for assault - non-sexual, non-DV during testing window",
            CCat12_4_During = "No. incidents charged for burglary during testing window",
            CCat12_5_During = "No. incidents charged for criminal sexual conduct during testing window",
            CCat12_6_During = "No. incidents charged for drug crime during testing window",
            CCat12_7_During = "No. incidents charged for homicide during testing window",
            CCat12_8_During = "No. incidents charged for larceny/theft/fraud during testing window",
            CCat12_9_During = "No. incidents charged for robbery during testing window",
            CCat12_10_During = "No. incidents charged for sex crime - other, excluding CSC during testing window",
            CCat12_11_During = "No. incidents charged for traffic/ordinances during testing window",
            CCat12_12_During = "No. incidents charged for weapons during testing window",
            CCat12_999_During = "No. incidents charged for excluded user-missing during testing window",
            CCat12_Any_During = "No. incidents charged for any of 12 main crime categories during testing window",
            CCat12_Sexual_During = "No. incidents charged for any sexual crimes (2 categories) during testing window",
            CCat12_Violent_During = "No. incidents charged for any violent crimes (5 categories) during testing window",
            CCat12_Property_During = "No. incidents charged for any property crimes (3 categories) during testing window",
            CCat12_Other_During = "No. incidents charged for any other crimes (2 categories) during testing window",
            CCat12_1_After = "No. incidents charged for arson after testing window",
            CCat12_2_After = "No. incidents charged for assault - DV, stalking after testing window",
            CCat12_3_After = "No. incidents charged for assault - non-sexual, non-DV after testing window",
            CCat12_4_After = "No. incidents charged for burglary after testing window",
            CCat12_5_After = "No. incidents charged for criminal sexual conduct after testing window",
            CCat12_6_After = "No. incidents charged for drug crime after testing window",
            CCat12_7_After = "No. incidents charged for homicide after testing window",
            CCat12_8_After = "No. incidents charged for larceny/theft/fraud after testing window",
            CCat12_9_After = "No. incidents charged for robbery after testing window",
            CCat12_10_After = "No. incidents charged for sex crime - other, excluding CSC after testing window",
            CCat12_11_After = "No. incidents charged for traffic/ordinances after testing window",
            CCat12_12_After = "No. incidents charged for weapons after testing window",
            CCat12_999_After = "No. incidents charged for excluded user-missing",
            CCat12_Any_After = "No. incidents charged for any of 12 main crime categories after testing window",
            CCat12_Sexual_After = "No. incidents charged for any sexual crimes (2 categories) after testing window",
            CCat12_Violent_After = "No. incidents charged for any violent crimes (5 categories) after testing window",
            CCat12_Property_After = "No. incidents charged for any property crimes (3 categories) after testing window",
            CCat12_Other_After = "No. incidents charged for any other crimes (2 categories) after testing window") ->
  IDNEW
```

6.13.3 Incidents With Convictions (Overall Plus Before, During, and After Testing Window)

For any given offender (identifed by an *OID* value), there can be multiple incidents (identified by *IID* values). Aggregating the dummy variables CCat12_1 to CCat12_12 and CCat12_999 by summing across incidents for each offender allows us to get overall incident counts showing how often the offender was convicted for various types of crime over the course of the entire (documented) criminal history. The aggregated variables merged into *IDNEW* retain the same varable names as the source dummy-coded variables.

```
# Aggregate JCat12 dummy codes to offender level incident counts.
INCEW %>%
  group_by(OID) %>%
  # Aggregate dummy codes by OID into incident count variables
  summarize(across(.cols = all_of(jvars), .fns = sum))  %>%
  # Merge dummy variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(jvars), .fns = replace_na, replace = 0)) %>%
  var_labels(JCat12_1 = "No. incidents convicted for arson",
             JCat12_2 = "No. incidents convicted for assault - DV, stalking",
             JCat12_3 = "No. incidents convicted for assault - non-sexual, non-DV",
             JCat12_4 = "No. incidents convicted for burglary",
             JCat12_5 = "No. incidents convicted for criminal sexual conduct",
             JCat12_6 = "No. incidents convicted for drug crime",
             JCat12_7 = "No. incidents convicted for homicide",
             JCat12_8 = "No. incidents convicted for larceny/theft/fraud",
             JCat12_9 = "No. incidents convicted for robbery",
             JCat12_10 = "No. incidents convicted for sex crime - other, excluding CSC",
             JCat12_11 = "No. incidents convicted for traffic/ordinances",
             JCat12_12 = "No. incidents convicted for weapons",
             JCat12_999 = "No. incidents convicted for excluded user-missing",
             JCat12_Any = "No. incidents convicted for any of 12 main crime categories",
```

```
            JCat12_Sexual = "No. incidents convicted for any sexual crimes (2 categories)",
            JCat12_Violent = "No. incidents convicted for any violent crimes (5 categories)",
            JCat12_Property = "No. incidents convicted for any property crimes (3 categories)",
            JCat12_Other = "No. incidents convicted for any other crimes (2 categories)") ->
  IDNEW
```

We also want to get separate counts for each crime category broken down by when the incident occurred relative to the earliest testing window (*IWhen*). This next chunk does that. The variable naming convention just adds suffixes to the previously defined variable names to indicate the period to which each count variable pertains. Because some offenders' adult CHRs did not overlap with the before and during periods, we have to selectively recode counts of zero to missing data when the observed duration of the period in question was zero years. Otherwise we risk mistaking what amounts to a structural zero (no opportunity to observe incidents because because we had zero years of that period represented in the adult CHR) with a sampling zero (we had the opportunity to observe incidents because we observed more than 0 years of the period in the adult CHR, but just did not see any in that period).

```
# Aggregate JCat12 dummy codes to offender level incident counts by IWhen.
INCEW %>%
  group_by(OID, IWhen, .drop = FALSE) %>%
  pivot_wider(names_from = IWhen, values_from = all_of(jvars),
              values_fill = 0) %>%
  group_by(OID) %>%
  # Aggregate dummy codes by OID into incident count variables
  summarize(across(.cols = all_of(jvarsbda), .fns = sum, na.rm = TRUE))  %>%
  # Merge count variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Set count variables with values of 0 to NA when that period was unobserved.
  mutate(across(.cols = all_of(jvarsb),
                .fns = ~ifelse(Years_Before == 0, yes = NA, no = .))) %>%
  mutate(across(.cols = all_of(jvarsd),
                .fns = ~ifelse(Years_During == 0, yes = NA, no = .))) %>%
  var_labels(JCat12_1_Before = "No. incidents convicted for arson before testing window before testing window",
             JCat12_2_Before = "No. incidents convicted for assault - DV, stalking before testing window",
             JCat12_3_Before = "No. incidents convicted for assault - non-sexual, non-DV before testing window",
             JCat12_4_Before = "No. incidents convicted for burglary before testing window",
             JCat12_5_Before = "No. incidents convicted for criminal sexual conduct before testing window",
             JCat12_6_Before = "No. incidents convicted for drug crime before testing window",
             JCat12_7_Before = "No. incidents convicted for homicide before testing window",
             JCat12_8_Before = "No. incidents convicted for larceny/theft/fraud before testing window",
             JCat12_9_Before = "No. incidents convicted for robbery before testing window",
             JCat12_10_Before = "No. incidents convicted for sex crime - other, excluding CSC before testing window",
             JCat12_11_Before = "No. incidents convicted for traffic/ordinances before testing window",
             JCat12_12_Before = "No. incidents convicted for weapons before testing window",
             JCat12_999_Before = "No. incidents convicted for excluded user-missing before testing window",
             JCat12_Any_Before = "No. incidents convicted for any of 12 main crime categories before testing window",
             JCat12_Sexual_Before = "No. incidents convicted for any sexual crimes (2 categories) before testing window",
             JCat12_Violent_Before = "No. incidents convicted for any violent crimes (5 categories) before testing window",
             JCat12_Property_Before = "No. incidents convicted for any property crimes (3 categories) before testing window",
             JCat12_Other_Before = "No. incidents convicted for any other crimes (2 categories) before testing window",
             JCat12_1_During = "No. incidents convicted for arson during testing window",
             JCat12_2_During = "No. incidents convicted for assault - DV, stalking during testing window",
             JCat12_3_During = "No. incidents convicted for assault - non-sexual, non-DV during testing window",
             JCat12_4_During = "No. incidents convicted for burglary during testing window",
             JCat12_5_During = "No. incidents convicted for criminal sexual conduct during testing window",
             JCat12_6_During = "No. incidents convicted for drug crime during testing window",
             JCat12_7_During = "No. incidents convicted for homicide during testing window",
             JCat12_8_During = "No. incidents convicted for larceny/theft/fraud during testing window",
             JCat12_9_During = "No. incidents convicted for robbery during testing window",
             JCat12_10_During = "No. incidents convicted for sex crime - other, excluding CSC during testing window",
             JCat12_11_During = "No. incidents convicted for traffic/ordinances during testing window",
             JCat12_12_During = "No. incidents convicted for weapons during testing window",
             JCat12_999_During = "No. incidents convicted for excluded user-missing during testing window",
             JCat12_Any_During = "No. incidents convicted for any of 12 main crime categories during testing window",
             JCat12_Sexual_During = "No. incidents convicted for any sexual crimes (2 categories) during testing window",
             JCat12_Violent_During = "No. incidents convicted for any violent crimes (5 categories) during testing window",
             JCat12_Property_During = "No. incidents convicted for any property crimes (3 categories) during testing window",
             JCat12_Other_During = "No. incidents convicted for any other crimes (2 categories) during testing window",
             JCat12_1_After = "No. incidents convicted for arson after testing window",
             JCat12_2_After = "No. incidents convicted for assault - DV, stalking after testing window",
             JCat12_3_After = "No. incidents convicted for assault - non-sexual, non-DV after testing window",
```

```
            JCat12_4_After = "No. incidents convicted for burglary after testing window",
            JCat12_5_After = "No. incidents convicted for criminal sexual conduct after testing window",
            JCat12_6_After = "No. incidents convicted for drug crime after testing window",
            JCat12_7_After = "No. incidents convicted for homicide after testing window",
            JCat12_8_After = "No. incidents convicted for larceny/theft/fraud after testing window",
            JCat12_9_After = "No. incidents convicted for robbery after testing window",
            JCat12_10_After = "No. incidents convicted for sex crime - other, excluding CSC after testing window",
            JCat12_11_After = "No. incidents convicted for traffic/ordinances after testing window",
            JCat12_12_After = "No. incidents convicted for weapons after testing window",
            JCat12_999_After = "No. incidents convicted for excluded user-missing",
            JCat12_Any_After = "No. incidents convicted for any of 12 main crime categories after testing window",
            JCat12_Sexual_After = "No. incidents convicted for any sexual crimes (2 categories) after testing window",
            JCat12_Violent_After = "No. incidents convicted for any violent crimes (5 categories) after testing window",
            JCat12_Property_After = "No. incidents convicted for any property crimes (3 categories) after testing window",
            JCat12_Other_After = "No. incidents convicted for any other crimes (2 categories) after testing window") ->
  IDNEW
```

6.13.4 Incidents with Any History (Arrest, Charge, or Conviction)

Having created the crime category history variables at the incident level, next we aggregate them to offender-level incident count varaiables named *HXCat12_1* to *HXCat12_12* and *HXCat12_999*. Each of these will contain the number of incidents that had a history showing the offender was *arrested for, charged with, or convicted of* the offense in question.

Aggregating the *INCEW* data on HXCat12_Any to offender-level record counts and merging them onto *IDNEW* effectively creates creates an alternate version of *NINC* called *HXCat12_Any* that only counts records associated with incidents where the offender was *arrested for, charged with, or convicted of* one or more of the 12 main crime categories.

```
INCEW %>%
  group_by(OID, .drop = FALSE) %>%
  # Aggregate dummy codes by OID into incident count variables
  summarize(across(.cols = all_of(hxvars), .fns = sum))  %>%
  # Merge count variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Replace all NA values in the new variables with 0.
  mutate(across(.cols = all_of(hxvars), .fns = replace_na, replace = 0)) %>%
  var_labels(HXCat12_1 = "No. incidents arrested, charged, or convicted for arson",
             HXCat12_2 = "No. incidents arrested, charged, or convicted for assault - DV, stalking",
             HXCat12_3 = "No. incidents arrested, charged, or convicted for assault - non-sexual, non-DV",
             HXCat12_4 = "No. incidents arrested, charged, or convicted for burglary",
             HXCat12_5 = "No. incidents arrested, charged, or convicted for criminal sexual conduct",
             HXCat12_6 = "No. incidents arrested, charged, or convicted for drug crime",
             HXCat12_7 = "No. incidents arrested, charged, or convicted for homicide",
             HXCat12_8 = "No. incidents arrested, charged, or convicted for larceny/theft/fraud",
             HXCat12_9 = "No. incidents arrested, charged, or convicted for robbery",
             HXCat12_10 = "No. incidents arrested, charged, or convicted for sex crime - other, excluding CSC",
             HXCat12_11 = "No. incidents arrested, charged, or convicted for traffic/ordinances",
             HXCat12_12 = "No. incidents arrested, charged, or convicted for weapons",
             HXCat12_999 = "No. incidents arrested, charged, or convicted for excluded user-missing",
             HXCat12_Any = "No. incidents arrested, charged, or convicted for any of 12 main crime categories",
             HXCat12_Sexual = "No. incidents arrested, charged, or convicted for any sexual crimes (2 categories)",
             HXCat12_Violent = "No. incidents arrested, charged, or convicted for any violent crimes (5 categories)",
             HXCat12_Property = "No. incidents arrested, charged, or convicted for any property crimes (3 categories)",
             HXCat12_Other = "No. incidents arrested, charged, or convicted for any other crimes (2 categories)") ->
  IDNEW
```

Now, we need to get offender-level breakdowns of those incident counts for incidents that occurred before, during, and after the testing window. We can aggregate as follows to get them and merge them into *IDNEW*.

This means that *HXCat12_Any_Before*, *HXCat12_Any_During*, and *HXCat12_Any_After* are respectively alternate versions of *NINC_Before*, *NINC_During*, and *NINC_After* that only count records associated with incidents where the offender was *arrested for, charged with, or convicted of* one or more of the 12 main crime categories.

Because some offenders' adult CHRs did not overlap with the before and during periods, we have to selectively recode counts of zero to missing data when the observed duration of the period in question was zero years. Otherwise we risk mistaking what amounts to a structural zero (no opportunity to observe incidents because because we had zero years of that period represented in the adult CHR) with a sampling zero (we had the opportunity to observe incidents because we observed more than 0 years of the period in the adult CHR, but just did not see any in that period).

```
# Aggregate HXCat12 dummy codes to offender level incident counts by IWhen.
INCEW %>%
  group_by(OID, IWhen, .drop = FALSE) %>%
  pivot_wider(names_from = IWhen, values_from = all_of(hxvars),
              values_fill = 0) %>%
  group_by(OID) %>%
  # Aggregate dummy codes by OID into incident count variables
  summarize(across(.cols = all_of(hxvarsbda), .fns = sum, na.rm = TRUE))  %>%
  # Merge count variables onto IDNEW.
  left_join(x = IDNEW, y = ., by = "OID") %>%
  # Set count variables with values of 0 to NA when that period was unobserved.
  mutate(across(.cols = all_of(hxvarsb),
                .fns = ~ifelse(Years_Before == 0, yes = NA, no = .))) %>%
  mutate(across(.cols = all_of(hxvarsd),
                .fns = ~ifelse(Years_During == 0, yes = NA, no = .))) %>%
  var_labels(HXCat12_1_Before = "No. incidents arrested, charged, or convicted for arson before testing window",
             HXCat12_2_Before = "No. incidents arrested, charged, or convicted for assault - DV, stalking before testing window",
             HXCat12_3_Before = "No. incidents arrested, charged, or convicted for assault - non-sexual, non-DV before testing window ",
             HXCat12_4_Before = "No. incidents arrested, charged, or convicted for burglary before testing window",
             HXCat12_5_Before = "No. incidents arrested, charged, or convicted for criminal sexual conduct before testing window",
             HXCat12_6_Before = "No. incidents arrested, charged, or convicted for drug crime before testing window",
             HXCat12_7_Before = "No. incidents arrested, charged, or convicted for homicide before testing window",
             HXCat12_8_Before = "No. incidents arrested, charged, or convicted for larceny/theft/fraud before testing window",
             HXCat12_9_Before = "No. incidents arrested, charged, or convicted for robbery before testing window",
             HXCat12_10_Before = "No. incidents arrested, charged, or convicted for sex crime - other, excluding CSC before testing wind
             HXCat12_11_Before = "No. incidents arrested, charged, or convicted for traffic/ordinances before testing window",
             HXCat12_12_Before = "No. incidents arrested, charged, or convicted for weapons before testing window",
             HXCat12_999_Before = "No. incidents arrested, charged, or convicted for excluded user-missing before testing window",
             HXCat12_Any_Before = "No. incidents arrested, charged, or convicted for any of 12 main crime categories before testing wind
             HXCat12_Sexual_Before = "No. incidents arrested, charged, or convicted for any sexual crimes (2 categories) before testing
             HXCat12_Violent_Before = "No. incidents arrested, charged, or convicted for any violent crimes (5 categories) before testin
             HXCat12_Property_Before = "No. incidents arrested, charged, or convicted for any property crimes (3 categories) before test
             HXCat12_Other_Before = "No. incidents arrested, charged, or convicted for any other crimes (2 categories) before testing wi
             HXCat12_1_During = "No. incidents arrested, charged, or convicted for arson during testing window",
             HXCat12_2_During = "No. incidents arrested, charged, or convicted for assault - DV, stalking during testing window",
             HXCat12_3_During = "No. incidents arrested, charged, or convicted for assault - non-sexual, non-DV during testing window",
             HXCat12_4_During = "No. incidents arrested, charged, or convicted for burglary during testing window",
             HXCat12_5_During = "No. incidents arrested, charged, or convicted for criminal sexual conduct during testing window",
             HXCat12_6_During = "No. incidents arrested, charged, or convicted for drug crime during testing window",
             HXCat12_7_During = "No. incidents arrested, charged, or convicted for homicide during testing window",
             HXCat12_8_During = "No. incidents arrested, charged, or convicted for larceny/theft/fraud during testing window",
             HXCat12_9_During = "No. incidents arrested, charged, or convicted for robbery during testing window",
             HXCat12_10_During = "No. incidents arrested, charged, or convicted for sex crime - other, excluding CSC during testing wind
             HXCat12_11_During = "No. incidents arrested, charged, or convicted for traffic/ordinances during testing window",
             HXCat12_12_During = "No. incidents arrested, charged, or convicted for weapons during testing window",
             HXCat12_999_During = "No. incidents arrested, charged, or convicted for excluded user-missing during testing window",
             HXCat12_Any_During = "No. incidents arrested, charged, or convicted for any of 12 main crime categories during testing wind
             HXCat12_Sexual_During = "No. incidents arrested, charged, or convicted for any sexual crimes (2 categories) during testing
             HXCat12_Violent_During = "No. incidents arrested, charged, or convicted for any violent crimes (5 categories) during testin
             HXCat12_Property_During = "No. incidents arrested, charged, or convicted for any property crimes (3 categories) during test
             HXCat12_Other_During = "No. incidents arrested, charged, or convicted for any other crimes (2 categories) during testing wi
             HXCat12_1_After = "No. incidents arrested, charged, or convicted for arson after testing window",
             HXCat12_2_After = "No. incidents arrested, charged, or convicted for assault - DV, stalking after testing window",
             HXCat12_3_After = "No. incidents arrested, charged, or convicted for assault - non-sexual, non-DV after testing window",
             HXCat12_4_After = "No. incidents arrested, charged, or convicted for burglary after testing window",
             HXCat12_5_After = "No. incidents arrested, charged, or convicted for criminal sexual conduct after testing window",
             HXCat12_6_After = "No. incidents arrested, charged, or convicted for drug crime after testing window",
             HXCat12_7_After = "No. incidents arrested, charged, or convicted for homicide after testing window",
             HXCat12_8_After = "No. incidents arrested, charged, or convicted for larceny/theft/fraud after testing window",
             HXCat12_9_After = "No. incidents arrested, charged, or convicted for robbery after testing window",
             HXCat12_10_After = "No. incidents arrested, charged, or convicted for sex crime - other, excluding CSC after testing window
             HXCat12_11_After = "No. incidents arrested, charged, or convicted for traffic/ordinances after testing window",
             HXCat12_12_After = "No. incidents arrested, charged, or convicted for weapons after testing window",
             HXCat12_999_After = "No. incidents arrested, charged, or convicted for excluded user-missing after testing window",
             HXCat12_Any_After = "No. incidents arrested, charged, or convicted for any of 12 main crime categories after testing window
             HXCat12_Sexual_After = "No. incidents arrested, charged, or convicted for any sexual crimes (2 categories) after testing wi
             HXCat12_Violent_After = "No. incidents arrested, charged, or convicted for any violent crimes (5 categories) after testing
             HXCat12_Property_After = "No. incidents arrested, charged, or convicted for any property crimes (3 categories) after testin
             HXCat12_Other_After = "No. incidents arrested, charged, or convicted for any other crimes (2 categories) after testing wind
  IDNEW
```

7 Compute Crime Category Count Variables in IDNEW

Here we compute the number of crime categories for which an offender was (a) arrested, (b) charged, (c) convicted, and (d) arrested, charged, or convicted. We store separate variables for overall counts, and counts broken down by when the incidents occurred relative to the testing window (*IWhen*).

```
IDNEW %>%
  # Count the number of crime categories associated w/ each offender.
  rowwise() %>%
  mutate(ACat12_Count = sum(c_across(all_of(avars[1:12])) > 0),
         ACat12_Count_Before = sum(c_across(all_of(avarsb[1:12])) > 0),
         ACat12_Count_During = sum(c_across(all_of(avarsd[1:12])) > 0),
         ACat12_Count_After = sum(c_across(all_of(avarsa[1:12])) > 0),
         CCat12_Count = sum(c_across(all_of(cvars[1:12])) > 0),
         CCat12_Count_Before = sum(c_across(all_of(cvarsb[1:12])) > 0),
         CCat12_Count_During = sum(c_across(all_of(cvarsd[1:12])) > 0),
         CCat12_Count_After = sum(c_across(all_of(cvarsa[1:12])) > 0),
         JCat12_Count = sum(c_across(all_of(jvars[1:12])) > 0),
         JCat12_Count_Before = sum(c_across(all_of(jvarsb[1:12])) > 0),
         JCat12_Count_During = sum(c_across(all_of(jvarsd[1:12])) > 0),
         JCat12_Count_After = sum(c_across(all_of(jvarsa[1:12])) > 0),
         HXCat12_Count = sum(c_across(all_of(hxvars[1:12])) > 0),
         HXCat12_Count_Before = sum(c_across(all_of(hxvarsb[1:12])) > 0),
         HXCat12_Count_During = sum(c_across(all_of(hxvarsd[1:12])) > 0),
         HXCat12_Count_After = sum(c_across(all_of(hxvarsa[1:12])) > 0)) %>%
  var_labels(ACat12_Count = "Arrested crime category count (overall)",
             ACat12_Count_Before = "Arrested crime category count (incidents before testing window)",
             ACat12_Count_During = "Arrested crime category count (incidents during testing window)",
             ACat12_Count_After = "Arrested crime category count (incidents after testing window)",
             CCat12_Count = "Charged crime category count (overall)",
             CCat12_Count_Before = "Charged crime category count (incidents before testing window)",
             CCat12_Count_During = "Charged crime category count (incidents during testing window)",
             CCat12_Count_After = "Charged crime category count (incidents after testing window)",
             JCat12_Count = "Convicted crime category count (overall)",
             JCat12_Count_Before = "Convicted crime category count (incidents before testing window)",
             JCat12_Count_During = "Convicted crime category count (incidents during testing window)",
             JCat12_Count_After = "Convicted crime category count (incidents after testing window)",
             HXCat12_Count = "Arrested, charged, or convicted crime category count (overall)",
             HXCat12_Count_Before = "Arrested, charged, or convicted crime category count (incidents before testing window)",
             HXCat12_Count_During = "Arrested, charged, or convicted crime category count (incidents during testing window)",
             HXCat12_Count_After = "Arrested, charged, or convicted crime category count (incidents after testing window)") ->
  IDNEW
```

8 Check Assumptions

We actually have few redundant incident count variables in *IDNEW* now. The variables OCSC1, OCSC2, and OCSC3 were created and stored in the SPSS file during the original grant work. The variables ACat12_5, CCat12_5, and JCat12_5 respectively contain exactly the same data, but were created above in the **Aggregating Crime Category Variables** section. They should agree with one another and we can check that as follows to prove that the aggregation logic was implemented the same way.

```
# Verify that OCSC1 = ACat12_5, OCSC2 = CCat12_5, & OCSC3 = JCat12_5.
all(IDNEW$OCSC1 == IDNEW$ACat12_5) # Arrest incident counts
```

```
## [1] TRUE
```

```
all(IDNEW$OCSC2 == IDNEW$CCat12_5) # Charge incident counts
```

```
## [1] TRUE
```

```
all(IDNEW$OCSC3 == IDNEW$JCat12_5) # Conviction incident counts
```

```
## [1] TRUE
```

9 Criminal History Record (CHR) Count Overview

Below is a summary of the record counts for different types of CHR records, along with the numbers of unique incidents and unique offenders represented in those records. It also shows the range of event dates associated with each record type (i.e., birth, incident, arrest, charge, and adjudication dates).

```
# Table caption.
TCap <- paste("Criminal History Record Counts and Date Ranges Before Versus",
              "After Case Selection")

# Summarize record counts before & after case selection for this study.
bind_rows(dfsummary(IDN, label = "Offenders...Before", dvar = IDN$DOB),
          dfsummary(IDNEW, label = "Offenders...After", dvar = IDNEW$DOB),
          dfsummary(INC, label = "Incidents...Before", dvar = INC$IDate),
          dfsummary(INCEW, label = "Incidents...After", dvar = INCEW$IDate),
          dfsummary(ARR, label = "Arrest offenses...Before", dvar = ARR$ADate),
          dfsummary(ARREW, label = "Arrest offenses...After",
                    dvar = ARREW$ADate),
          dfsummary(CHG, label = "Prosecutor charges...Before",
                    dvar = CHG$CDate),
          dfsummary(CHGEW, label = "Prosecutor charges...After",
                    dvar = CHGEW$CDate),
          dfsummary(JUD, label = "Adjudicated charges (all dispositions)...Before",
                    dvar = JUD$JDate),
          dfsummary(JUDEW, label = "Adjudicated charges (all dispositions)...After",
                    dvar = JUDEW$JDate),
          dfsummary(CON, label = "Adjudicated charges (convictions)...Before",
                    dvar = CON$JDate),
          dfsummary(CONEW, label = "Adjudicated charges (convictions)...After",
                    dvar = CONEW$JDate)) ->
  CHR.RC
kable(CHR.RC, format = "latex", booktabs = TRUE, caption = TCap,
      format.args = list(big.mark = ','))
```

	N.Records	N.Incidents	N.Offenders	Earliest	Latest
Offenders...Before	1,142	0	1,142	1938-05-30	1993-06-05
Offenders...After	1,082	0	1,082	1938-05-30	1993-06-05
Incidents...Before	9,550	9,550	1,142	1965-02-17	2016-04-11
Incidents...After	8,588	8,588	1,082	1965-02-17	2016-04-11
Arrest offenses...Before	9,826	9,013	1,142	1965-02-17	2016-04-11
Arrest offenses...After	8,945	8,242	1,082	1965-02-17	2016-04-11
Prosecutor charges...Before	6,052	4,876	1,135	1987-07-30	2016-04-06
Prosecutor charges...After	5,632	4,518	1,073	1987-07-30	2016-04-06
Adjudicated charges (all dispositions)...Before	12,522	6,642	1,131	1966-11-28	2016-04-11
Adjudicated charges (all dispositions)...After	10,995	5,740	1,070	1966-11-28	2016-04-11
Adjudicated charges (convictions)...Before	6,971	5,018	1,115	1966-11-28	2016-03-28
Adjudicated charges (convictions)...After	6,021	4,324	1,053	1966-11-28	2016-03-28

Table 11: Criminal History Record Counts and Date Ranges Before Versus After Case Selection

10 Create A Long Person-Period Version of IDNEW

The chunk below reorganizes the *IDNEW* data to a long person-period format with three rows per offender (one row for each period before, during, and after the testing window) and stores it as *IDNEWL*. The period (before, during, or after the earliest testing window) is identified by the *When* variable. The observed duration of the period is stored in *Years*, so *Years* > 0 indicates that the period overlapped offender's observed adult CHR period for at least one day. Meanwhile, *Years* $= 0$ indicates an unobserved period that did not overlap with the offender's observed adult CHR period at all. We will use *IDNEWL* later for running GEE models.

```
IDNEW %>%
  select(OID, Years_Before, Years_During, Years_After,
         HXCat12_Any_Before, HXCat12_Any_During, HXCat12_Any_After,
         HXCat12_Sexual_Before, HXCat12_Sexual_During, HXCat12_Sexual_After,
         HXCat12_Violent_Before, HXCat12_Violent_During, HXCat12_Violent_After,
         HXCat12_Property_Before, HXCat12_Property_During, HXCat12_Property_After,
         HXCat12_Other_Before, HXCat12_Other_During, HXCat12_Other_After,
         HXCat12_Count_Before, HXCat12_Count_During, HXCat12_Count_After,
         HXCat12_1_Before, HXCat12_1_During, HXCat12_1_After,
         HXCat12_2_Before, HXCat12_2_During, HXCat12_2_After,
         HXCat12_3_Before, HXCat12_3_During, HXCat12_3_After,
         HXCat12_4_Before, HXCat12_4_During, HXCat12_4_After,
         HXCat12_5_Before, HXCat12_5_During, HXCat12_5_After,
         HXCat12_6_Before, HXCat12_6_During, HXCat12_6_After,
         HXCat12_7_Before, HXCat12_7_During, HXCat12_7_After,
         HXCat12_8_Before, HXCat12_8_During, HXCat12_8_After,
         HXCat12_9_Before, HXCat12_9_During, HXCat12_9_After,
         HXCat12_10_Before, HXCat12_10_During, HXCat12_10_After,
         HXCat12_11_Before, HXCat12_11_During, HXCat12_11_After,
         HXCat12_12_Before, HXCat12_12_During, HXCat12_12_After) %>%
  pivot_longer(cols = -OID, names_to = c(".value", "When"),
               names_pattern = "(.*)_(.*)",
               values_to = c("Duration", "HXCat12_Any", "HXCat12_Sexual",
                             "HXCat12_Violent", "HXCat12_Property",
                             "HXCat12_Other", "HXCat12_Count",
                             "HXCat12_1", "HXCat12_2", "HXCat12_3",
                             "HXCat12_4", "HXCat12_5", "HXCat12_6",
                             "HXCat12_7", "HXCat12_8", "HXCat12_9",
                             "HXCat12_10", "HXCat12_11", "HXCat12_12")) %>%
  #rename(NINC12 = HXCat12_Any) %>%
  mutate(When = factor(x = When, levels = c("After", "Before", "During"))) %>%
  var_labels(When = "When period occurred (relative to testing window)",
             Years = "Years of period duration",
             HXCat12_Any = "No. incidents arrested, charged, or convicted for any of 12 main crime categories during period",
             HXCat12_Sexual = "No. incidents arrested, charged, or convicted for any sexual crimes (2 categories) during period",
             HXCat12_Violent = "No. incidents arrested, charged, or convicted for any violent crimes (5 categories) during period",
             HXCat12_Property = "No. incidents arrested, charged, or convicted for any property crimes (3 categories) during period",
             HXCat12_Other = "No. incidents arrested, charged, or convicted for any other crimes (2 categories) during period",
             HXCat12_Count = "Arrested, charged, or convicted crime category count (incidents during period)",
             HXCat12_1 = "No. incidents arrested, charged, or convicted for arson during period",
             HXCat12_2 = "No. incidents arrested, charged, or convicted for assault - DV, stalking during period",
             HXCat12_3 = "No. incidents arrested, charged, or convicted for assault - non-sexual, non-DV during period",
             HXCat12_4 = "No. incidents arrested, charged, or convicted for burglary during period",
             HXCat12_5 = "No. incidents arrested, charged, or convicted for criminal sexual conduct during period",
             HXCat12_6 = "No. incidents arrested, charged, or convicted for drug crime during period",
             HXCat12_7 = "No. incidents arrested, charged, or convicted for homicide during period",
             HXCat12_8 = "No. incidents arrested, charged, or convicted for larceny/theft/fraud during period",
             HXCat12_9 = "No. incidents arrested, charged, or convicted for robbery during period",
             HXCat12_10 = "No. incidents arrested, charged, or convicted for sex crime - other, excluding CSC during period",
             HXCat12_11 = "No. incidents arrested, charged, or convicted for traffic/ordinances during period",
             HXCat12_12 = "No. incidents arrested, charged, or convicted for weapons during period") ->
  IDNEWL
```

11 Extract the After Period Records from IDNEWL to IDNEWA

The chunk below filters the *IDNEWL* tibble down to just the records for the after period (*When* = "After"), then reorganizes it to a long format with one row per offender per variable to facilitate analyses we added to respond to peer review feedback on the manuscript. We store the resulting data in *IDNEWA*.

```
# Create objects to hold vectors of variable names that should be grouped
# together in output later.
Any_Crimes      <- c("HXCat12_Any")
Sexual_Crimes   <- c("HXCat12_5", "HXCat12_10", "HXCat12_Sexual")
Violent_Crimes  <- c("HXCat12_2", "HXCat12_3", "HXCat12_7", "HXCat12_9",
                     "HXCat12_12", "HXCat12_Violent")
Property_Crimes <- c("HXCat12_1", "HXCat12_4", "HXCat12_8", "HXCat12_Property")
Other_Crimes    <- c("HXCat12_6", "HXCat12_11", "HXCat12_Other")
```

```
# Create vector of labels for the 12 main crime categories.
CCLabels <- c("Criminal Sexual Conduct (CSC)",
              "Sex Crimes (non-CSC crimes)",
              "Assault, Domestic Violence",
              "Assault, non-Domestic Violence",
              "Homicide",
              "Robbery",
              "Weapons",
              "Arson",
              "Burglary",
              "Larceny, Theft, Fraud",
              "Drug Crimes",
              "Traffic & Ordinances")

# Create vector of labels for the 5 broader crime categories.
BCLabels <- c("Any Crimes (12 categories)",
              "Sexual Crimes (2 categories)",
              "Violent Non-Sexual Crimes (5 categories)",
              "Property Crimes (3 categories)",
              "Other Crimes (2 categories)")

# Create the new tibble.
IDNEWL %>%
  filter(When == "After") %>%
  pivot_longer(data = ., cols = starts_with("HXCat12_"), names_to = "Variable",
               values_to = "Count") %>%
  mutate(# Create a variable to store variable labels we will need for tables & plots.
         VLabel = case_when(Variable == "HXCat12_Any" ~ "Any Crimes (12 categories)",
                            Variable == "HXCat12_Sexual" ~ "Sexual Crimes (2 categories)",
                            Variable == "HXCat12_Violent" ~ "Violent Non-Sexual Crimes (5 categories)",
                            Variable == "HXCat12_Property" ~ "Property Crimes (3 categories)",
                            Variable == "HXCat12_Other" ~ "Other Crimes (2 categories)",
                            Variable == "HXCat12_Count" ~ "Crime Category",
                            Variable == "HXCat12_1" ~ "Arson",
                            Variable == "HXCat12_2" ~ "Assault, Domestic Violence",
                            Variable == "HXCat12_3" ~ "Assault, non-Domestic Violence",
                            Variable == "HXCat12_4" ~ "Burglary",
                            Variable == "HXCat12_5" ~ "Criminal Sexual Conduct (CSC)",
                            Variable == "HXCat12_6" ~ "Drug Crimes",
                            Variable == "HXCat12_7" ~ "Homicide",
                            Variable == "HXCat12_8" ~ "Larceny, Theft, Fraud",
                            Variable == "HXCat12_9" ~ "Robbery",
                            Variable == "HXCat12_10" ~ "Sex Crimes (non-CSC crimes)",
                            Variable == "HXCat12_11" ~ "Traffic & Ordinances",
                            Variable == "HXCat12_12" ~ "Weapons"),
         # Add a variable to group rows into subsets we may need later.
         VGroup = case_when(Variable == "HXCat12_Count" ~ "Crime Category Count",
                            Variable %in% Any_Crimes ~ "Any Crimes",
                            Variable %in% Sexual_Crimes ~ "Sexual Crimes",
                            Variable %in% Violent_Crimes ~ "Violent Non-Sexual Crimes",
                            Variable %in% Property_Crimes ~ "Property Crimes",
                            Variable %in% Other_Crimes ~ "Other Crimes"),
         # Create a variable for the sort order of rows.
         VOrder = case_when(Variable == "HXCat12_Count" ~ 0,
                            Variable == "HXCat12_Any" ~ 1,
                            Variable == "HXCat12_Sexual" ~ 2,
                            Variable == "HXCat12_5" ~ 3,
                            Variable == "HXCat12_10" ~ 4,
                            Variable == "HXCat12_Violent" ~ 5,
                            Variable == "HXCat12_2" ~ 6,
                            Variable == "HXCat12_3" ~ 7,
                            Variable == "HXCat12_7" ~ 8,
                            Variable == "HXCat12_9" ~ 9,
                            Variable == "HXCat12_12" ~ 10,
                            Variable == "HXCat12_Property" ~ 11,
                            Variable == "HXCat12_1" ~ 12,
                            Variable == "HXCat12_4" ~ 13,
                            Variable == "HXCat12_8" ~ 14,
                            Variable == "HXCat12_Other" ~ 15,
                            Variable == "HXCat12_6" ~ 16,
                            Variable == "HXCat12_11" ~ 17),
```

```
        # Create a factor variable for the 12 main crime categories.
        Crime = factor(VLabel, levels = CCLabels, labels = CCLabels),
        # Create a factor variable for the 5 broader crime categories.
        BCrime = factor(VLabel, levels = BCLabels, labels = BCLabels)) %>%
  arrange(OID, VOrder) %>%
  var_labels(Count = "No. incidents arrested, charged, or convicted after the testing window",
             Variable = "Variable name",
             VLabel = "Variable label",
             VGroup = "Variable group",
             VOrder = "Variable sorting order",
             Crime = "Crime category (12 levels)",
             BCrime = "Broader crime category (4 levels)") ->
  IDNEWA
```

12 Extract the Before Period Records from IDNEWL to IDNEWB

The chunk below filters the *IDNEWL* tibble down to just the records for the after period (*When* = "Before"), then reorganizes it to a long format with one row per offender per variable to facilitate analyses we added to respond to peer review feedback on the manuscript. We store the resulting data in *IDNEWB*.

```
# Create the new tibble.
IDNEWL %>%
  filter(When == "Before") %>%
  pivot_longer(data = ., cols = starts_with("HXCat12_"), names_to = "Variable",
               values_to = "Count") %>%
  mutate(# Create a variable to store variable labels we will need for tables & plots.
         VLabel = case_when(Variable == "HXCat12_Any" ~ "Any Crimes (12 categories)",
                            Variable == "HXCat12_Sexual" ~ "Sexual Crimes (2 categories)",
                            Variable == "HXCat12_Violent" ~ "Violent Non-Sexual Crimes (5 categories)",
                            Variable == "HXCat12_Property" ~ "Property Crimes (3 categories)",
                            Variable == "HXCat12_Other" ~ "Other Crimes (2 categories)",
                            Variable == "HXCat12_Count" ~ "Crime Category",
                            Variable == "HXCat12_1" ~ "Arson",
                            Variable == "HXCat12_2" ~ "Assault, Domestic Violence",
                            Variable == "HXCat12_3" ~ "Assault, non-Domestic Violence",
                            Variable == "HXCat12_4" ~ "Burglary",
                            Variable == "HXCat12_5" ~ "Criminal Sexual Conduct (CSC)",
                            Variable == "HXCat12_6" ~ "Drug Crimes",
                            Variable == "HXCat12_7" ~ "Homicide",
                            Variable == "HXCat12_8" ~ "Larceny, Theft, Fraud",
                            Variable == "HXCat12_9" ~ "Robbery",
                            Variable == "HXCat12_10" ~ "Sex Crimes (non-CSC crimes)",
                            Variable == "HXCat12_11" ~ "Traffic & Ordinances",
                            Variable == "HXCat12_12" ~ "Weapons"),
         # Add a variable to group rows into subsets we may need later.
         VGroup = case_when(Variable == "HXCat12_Count" ~ "Crime Category Count",
                            Variable %in% Any_Crimes ~ "Any Crimes",
                            Variable %in% Sexual_Crimes ~ "Sexual Crimes",
                            Variable %in% Violent_Crimes ~ "Violent Non-Sexual Crimes",
                            Variable %in% Property_Crimes ~ "Property Crimes",
                            Variable %in% Other_Crimes ~ "Other Crimes"),
         # Create a variable for the sort order of rows.
         VOrder = case_when(Variable == "HXCat12_Count" ~ 0,
                            Variable == "HXCat12_Any" ~ 1,
                            Variable == "HXCat12_Sexual" ~ 2,
                            Variable == "HXCat12_5" ~ 3,
                            Variable == "HXCat12_10" ~ 4,
                            Variable == "HXCat12_Violent" ~ 5,
                            Variable == "HXCat12_2" ~ 6,
                            Variable == "HXCat12_3" ~ 7,
                            Variable == "HXCat12_7" ~ 8,
                            Variable == "HXCat12_9" ~ 9,
                            Variable == "HXCat12_12" ~ 10,
                            Variable == "HXCat12_Property" ~ 11,
                            Variable == "HXCat12_1" ~ 12,
                            Variable == "HXCat12_4" ~ 13,
                            Variable == "HXCat12_8" ~ 14,
                            Variable == "HXCat12_Other" ~ 15,
```

```
                          Variable == "HXCat12_6" ~ 16,
                          Variable == "HXCat12_11" ~ 17),
         # Create a factor variable for the 12 main crime categories.
         Crime = factor(VLabel, levels = CCLabels, labels = CCLabels),
         # Create a factor variable for the 5 broader crime categories.
         BCrime = factor(VLabel, levels = BCLabels, labels = BCLabels)) %>%
  arrange(OID, VOrder) %>%
  var_labels(Count = "No. incidents arrested, charged, or convicted before the testing window",
             Variable = "Variable name",
             VLabel = "Variable label",
             VGroup = "Variable group",
             VOrder = "Variable sorting order",
             Crime = "Crime category (12 levels)",
             BCrime = "Broader crime category (4 levels)") ->
  IDNEWB
```

13 Extract the During Period Records from IDNEWL to IDNEWD

The chunk below filters the *IDNEWL* tibble down to just the records for the after period (*When* = "During"), then reorganizes it to a long format with one row per offender per variable to facilitate analyses we added to respond to peer review feedback on the manuscript. We store the resulting data in *IDNEWD*.

```
# Create the new tibble.
IDNEWL %>%
  filter(When == "During") %>%
  pivot_longer(data = ., cols = starts_with("HXCat12_"), names_to = "Variable",
               values_to = "Count") %>%
  mutate(# Create a variable to store variable labels we will need for tables & plots.
         VLabel = case_when(Variable == "HXCat12_Any" ~ "Any Crimes (12 categories)",
                            Variable == "HXCat12_Sexual" ~ "Sexual Crimes (2 categories)",
                            Variable == "HXCat12_Violent" ~ "Violent Non-Sexual Crimes (5 categories)",
                            Variable == "HXCat12_Property" ~ "Property Crimes (3 categories)",
                            Variable == "HXCat12_Other" ~ "Other Crimes (2 categories)",
                            Variable == "HXCat12_Count" ~ "Crime Category",
                            Variable == "HXCat12_1" ~ "Arson",
                            Variable == "HXCat12_2" ~ "Assault, Domestic Violence",
                            Variable == "HXCat12_3" ~ "Assault, non-Domestic Violence",
                            Variable == "HXCat12_4" ~ "Burglary",
                            Variable == "HXCat12_5" ~ "Criminal Sexual Conduct (CSC)",
                            Variable == "HXCat12_6" ~ "Drug Crimes",
                            Variable == "HXCat12_7" ~ "Homicide",
                            Variable == "HXCat12_8" ~ "Larceny, Theft, Fraud",
                            Variable == "HXCat12_9" ~ "Robbery",
                            Variable == "HXCat12_10" ~ "Sex Crimes (non-CSC crimes)",
                            Variable == "HXCat12_11" ~ "Traffic & Ordinances",
                            Variable == "HXCat12_12" ~ "Weapons"),
         # Add a variable to group rows into subsets we may need later.
         VGroup = case_when(Variable == "HXCat12_Count" ~ "Crime Category Count",
                            Variable %in% Any_Crimes ~ "Any Crimes",
                            Variable %in% Sexual_Crimes ~ "Sexual Crimes",
                            Variable %in% Violent_Crimes ~ "Violent Non-Sexual Crimes",
                            Variable %in% Property_Crimes ~ "Property Crimes",
                            Variable %in% Other_Crimes ~ "Other Crimes"),
         # Create a variable for the sort order of rows.
         VOrder = case_when(Variable == "HXCat12_Count" ~ 0,
                            Variable == "HXCat12_Any" ~ 1,
                            Variable == "HXCat12_Sexual" ~ 2,
                            Variable == "HXCat12_5" ~ 3,
                            Variable == "HXCat12_10" ~ 4,
                            Variable == "HXCat12_Violent" ~ 5,
                            Variable == "HXCat12_2" ~ 6,
                            Variable == "HXCat12_3" ~ 7,
                            Variable == "HXCat12_7" ~ 8,
                            Variable == "HXCat12_9" ~ 9,
                            Variable == "HXCat12_12" ~ 10,
                            Variable == "HXCat12_Property" ~ 11,
                            Variable == "HXCat12_1" ~ 12,
                            Variable == "HXCat12_4" ~ 13,
```

```
                          Variable == "HXCat12_8" ~ 14,
                          Variable == "HXCat12_Other" ~ 15,
                          Variable == "HXCat12_6" ~ 16,
                          Variable == "HXCat12_11" ~ 17),
        # Create a factor variable for the 12 main crime categories.
        Crime = factor(VLabel, levels = CCLabels, labels = CCLabels),
        # Create a factor variable for the 5 broader crime categories.
        BCrime = factor(VLabel, levels = BCLabels, labels = BCLabels)) %>%
  arrange(OID, VOrder) %>%
  var_labels(Count = "No. incidents arrested, charged, or convicted during the testing window",
             Variable = "Variable name",
             VLabel = "Variable label",
             VGroup = "Variable group",
             VOrder = "Variable sorting order",
             Crime = "Crime category (12 levels)",
             BCrime = "Broader crime category (4 levels)") ->
  IDNEWD
```

14 Extract the After Period Records from INCEW to INCEWA

The chunk below filters the *INCEW* tibble down to just the records for the after period (*When* = "After"), then reorganizes it to a long format with one row per offender per variable to facilitate analyses we added to respond to peer review feedback on the manuscript. We store the resulting data in *INCEWA*.

```
# Create the new tibble.
INCEW %>%
  filter(IWhen == "After") %>%
  pivot_longer(data = ., cols = starts_with("HXCat12_"), names_to = "Variable",
               values_to = "Include") %>%
  mutate(# Create a variable to store variable labels we will need for tables & plots.
        VLabel = case_when(Variable == "HXCat12_Any" ~ "Any Crimes (12 categories)",
                          Variable == "HXCat12_Sexual" ~ "Sexual Crimes (2 categories)",
                          Variable == "HXCat12_Violent" ~ "Violent Non-Sexual Crimes (5 categories)",
                          Variable == "HXCat12_Property" ~ "Property Crimes (3 categories)",
                          Variable == "HXCat12_Other" ~ "Other Crimes (2 categories)",
                          Variable == "HXCat12_Count" ~ "Crime Category",
                          Variable == "HXCat12_1" ~ "Arson",
                          Variable == "HXCat12_2" ~ "Assault, Domestic Violence",
                          Variable == "HXCat12_3" ~ "Assault, non-Domestic Violence",
                          Variable == "HXCat12_4" ~ "Burglary",
                          Variable == "HXCat12_5" ~ "Criminal Sexual Conduct (CSC)",
                          Variable == "HXCat12_6" ~ "Drug Crimes",
                          Variable == "HXCat12_7" ~ "Homicide",
                          Variable == "HXCat12_8" ~ "Larceny, Theft, Fraud",
                          Variable == "HXCat12_9" ~ "Robbery",
                          Variable == "HXCat12_10" ~ "Sex Crimes (non-CSC crimes)",
                          Variable == "HXCat12_11" ~ "Traffic & Ordinances",
                          Variable == "HXCat12_12" ~ "Weapons"),
        # Add a variable to group rows into subsets we may need later.
        VGroup = case_when(Variable == "HXCat12_Count" ~ "Crime Category Count",
                          Variable %in% Any_Crimes ~ "Any Crimes",
                          Variable %in% Sexual_Crimes ~ "Sexual Crimes",
                          Variable %in% Violent_Crimes ~ "Violent Non-Sexual Crimes",
                          Variable %in% Property_Crimes ~ "Property Crimes",
                          Variable %in% Other_Crimes ~ "Other Crimes"),
        # Create a variable for the sort order of rows.
        VOrder = case_when(Variable == "HXCat12_Count" ~ 0,
                          Variable == "HXCat12_Any" ~ 1,
                          Variable == "HXCat12_Sexual" ~ 2,
                          Variable == "HXCat12_5" ~ 3,
                          Variable == "HXCat12_10" ~ 4,
                          Variable == "HXCat12_Violent" ~ 5,
                          Variable == "HXCat12_2" ~ 6,
                          Variable == "HXCat12_3" ~ 7,
                          Variable == "HXCat12_7" ~ 8,
                          Variable == "HXCat12_9" ~ 9,
                          Variable == "HXCat12_12" ~ 10,
                          Variable == "HXCat12_Property" ~ 11,
```

```
                          Variable == "HXCat12_1" ~ 12,
                          Variable == "HXCat12_4" ~ 13,
                          Variable == "HXCat12_8" ~ 14,
                          Variable == "HXCat12_Other" ~ 15,
                          Variable == "HXCat12_6" ~ 16,
                          Variable == "HXCat12_11" ~ 17),
        # Create a factor variable for the 12 main crime categories.
        Crime = factor(VLabel, levels = CCLabels, labels = CCLabels),
        # Create a factor variable for the 5 broader crime categories.
        BCrime = factor(VLabel, levels = BCLabels, labels = BCLabels)) %>%
  filter(Include == 1) %>%
  arrange(OID, VOrder) %>%
  var_labels(Include = "Incident arrested, charged, or convicted after the testing window",
             Variable = "Variable name",
             VLabel = "Variable label",
             VGroup = "Variable group",
             VOrder = "Variable sorting order",
             Crime = "Crime category (12 levels)",
             BCrime = "Broader crime category (4 levels)") ->
  INCEWA
```

15 Save Data to a File

```
save(IDNEW, INCEW, ARREW, CHGEW, JUDEW, CONEW, IDNEWL, IDNEWA, IDNEWB, IDNEWD,
     INCEWA, file = here::here("data/CHR_Data.RData"))
```

16 Wrap Up

16.1 Project Information

These materials are scholarly products based on research funded by the following grant.

Campbell, R., Pierce, S. J., & Sharma, D. (2015–2018). *Serial sexual assaults: A longitudinal examination of offending patterns using DNA evidence.* (NIJ Award # 2014-NE-BX-0006) [Grant]. National Institute of Justice.

16.2 References

Campbell, R. (2019). *Serial sexual assaults: A longitudinal examination of offending patterns using DNA evidence, Detroit, Michigan, 2009* [Data files, codebooks, computer programs, and statistical output]. ICPSR37134-v1. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [distributor], 2019-02-28. Retrieved from: https://doi.org/10.3886/ICPSR37134.v1

16.3 Software Information

We use R Markdown to enhance reproducibility. Knitting the source R Markdown script *Step_01_Data_Mgt.Rmd* generates this PDF file containing explanatory text, R code, plus R output (text and graphics).

- We used RStudio to work with R and R markdown files. The software chain looks like this: **Rmd file > RStudio > R > rmarkdown > knitr > md file > pandoc > tex file > TinyTeX > PDF file**.
- We recommend using TinyTeX to compile LaTeX files into PDF files. However, it should be viable to use MiKTeX or another LaTeX distribution instead.
- We used pandoc 2.14.0.3 for this document.

This document was generated using the following computational environment and dependencies:

```
# Check and report whether we used TinyTex or other LaTeX software.
which_latex()
```

```
## [1] "is_tinytex = TRUE. We used TinyTeX."
```

```
# Get R and R package version numbers in use.
devtools::session_info()
```

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.1.2 (2021-11-01)
##  os       Windows 10 x64 (build 19042)
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  ctype    English_United States.1252
##  tz       America/New_York
##  date     2022-01-15
##  pandoc   2.14.0.3 @ C:/Program Files/RStudio/bin/pandoc/ (via rmarkdown)
##
## - Packages -------------------------------------------------------------------
##  package              * version date (UTC) lib source
##  abind                  1.4-5   2016-07-21 [1] CRAN (R 4.1.0)
##  assertive.base         0.0-9   2021-02-08 [1] CRAN (R 4.1.0)
##  assertive.properties   0.0-4   2016-12-30 [1] CRAN (R 4.1.0)
##  assertive.types        0.0-3   2016-12-30 [1] CRAN (R 4.1.0)
##  assertthat             0.2.1   2019-03-21 [1] CRAN (R 4.1.0)
##  cachem                 1.0.6   2021-08-19 [1] CRAN (R 4.1.1)
##  callr                  3.7.0   2021-04-20 [1] CRAN (R 4.1.0)
##  car                  * 3.0-12  2021-11-06 [1] CRAN (R 4.1.2)
##  carData              * 3.0-5   2022-01-06 [1] CRAN (R 4.1.2)
##  cli                    3.1.0   2021-10-27 [1] CRAN (R 4.1.1)
##  codetools              0.2-18  2020-11-04 [2] CRAN (R 4.1.2)
##  colorspace             2.0-2   2021-06-24 [1] CRAN (R 4.1.0)
##  crayon                 1.4.2   2021-10-29 [1] CRAN (R 4.1.1)
##  data.table             1.14.2  2021-09-27 [1] CRAN (R 4.1.1)
##  DBI                    1.1.2   2021-12-20 [1] CRAN (R 4.1.2)
##  desc                   1.4.0   2021-09-28 [1] CRAN (R 4.1.1)
##  descr                * 1.1.5   2021-02-16 [1] CRAN (R 4.1.0)
##  devtools               2.4.3   2021-11-30 [1] CRAN (R 4.1.2)
##  digest                 0.6.29  2021-12-01 [1] CRAN (R 4.1.2)
##  dplyr                * 1.0.7   2021-06-18 [1] CRAN (R 4.1.0)
##  ellipsis               0.3.2   2021-04-29 [1] CRAN (R 4.1.0)
##  evaluate               0.14    2019-05-28 [1] CRAN (R 4.1.0)
##  fansi                  0.5.0   2021-05-25 [1] CRAN (R 4.1.2)
##  farver                 2.1.0   2021-02-28 [1] CRAN (R 4.1.0)
##  fastmap                1.1.0   2021-01-25 [1] CRAN (R 4.1.0)
##  forcats                0.5.1   2021-01-27 [1] CRAN (R 4.1.0)
##  fs                     1.5.2   2021-12-08 [1] CRAN (R 4.1.2)
##  generics               0.1.1   2021-10-25 [1] CRAN (R 4.1.1)
##  ggplot2              * 3.3.5   2021-06-25 [1] CRAN (R 4.1.0)
##  ggrepel                0.9.1   2021-01-15 [1] CRAN (R 4.1.0)
##  git2r                  0.29.0  2021-11-22 [1] CRAN (R 4.1.2)
##  glue                   1.6.0   2021-12-17 [1] CRAN (R 4.1.2)
##  gtable                 0.3.0   2019-03-25 [1] CRAN (R 4.1.0)
##  haven                * 2.4.3   2021-08-04 [1] CRAN (R 4.1.0)
##  here                 * 1.0.1   2020-12-13 [1] CRAN (R 4.1.0)
##  hms                    1.1.1   2021-09-26 [1] CRAN (R 4.1.1)
##  htmltools              0.5.2   2021-08-25 [1] CRAN (R 4.1.1)
##  htmlwidgets            1.5.4   2021-09-08 [1] CRAN (R 4.1.1)
##  httr                   1.4.2   2020-07-20 [1] CRAN (R 4.1.0)
##  insight                0.15.0  2022-01-07 [1] CRAN (R 4.1.2)
##  jsonlite               1.7.2   2020-12-09 [1] CRAN (R 4.1.0)
##  kableExtra           * 1.3.4   2021-02-20 [1] CRAN (R 4.1.0)
##  knitr                * 1.37    2021-12-16 [1] CRAN (R 4.1.2)
##  lattice              * 0.20-45 2021-09-22 [1] CRAN (R 4.1.1)
##  lazyeval               0.2.2   2019-03-15 [1] CRAN (R 4.1.0)
##  lifecycle              1.0.1   2021-09-24 [1] CRAN (R 4.1.1)
##  lubridate            * 1.8.0   2021-10-07 [1] CRAN (R 4.1.1)
```

```
##  magrittr              2.0.1   2020-11-17 [1] CRAN (R 4.1.0)
##  memoise               2.0.1   2021-11-26 [1] CRAN (R 4.1.2)
##  mnormt                2.0.2   2020-09-01 [1] CRAN (R 4.1.0)
##  munsell               0.5.0   2018-06-12 [1] CRAN (R 4.1.0)
##  nlme                  3.1-153 2021-09-07 [1] CRAN (R 4.1.1)
##  pillar                1.6.4   2021-10-18 [1] CRAN (R 4.1.1)
##  pkgbuild              1.3.1   2021-12-20 [1] CRAN (R 4.1.2)
##  pkgconfig             2.0.3   2019-09-22 [1] CRAN (R 4.1.0)
##  pkgload               1.2.4   2021-11-30 [1] CRAN (R 4.1.2)
##  plotly                4.10.0  2021-10-09 [1] CRAN (R 4.1.1)
##  plyr                * 1.8.6   2020-03-03 [1] CRAN (R 4.1.0)
##  prettyunits           1.1.1   2020-01-24 [1] CRAN (R 4.1.0)
##  processx              3.5.2   2021-04-30 [1] CRAN (R 4.1.0)
##  ps                    1.6.0   2021-02-28 [1] CRAN (R 4.1.0)
##  psych               * 2.1.9   2021-09-22 [1] CRAN (R 4.1.1)
##  purrr                 0.3.4   2020-04-17 [1] CRAN (R 4.1.0)
##  R6                    2.5.1   2021-08-19 [1] CRAN (R 4.1.1)
##  RColorBrewer        * 1.1-2   2014-12-07 [1] CRAN (R 4.1.0)
##  Rcpp                  1.0.7   2021-07-07 [1] CRAN (R 4.1.0)
##  readr                 2.1.1   2021-11-30 [1] CRAN (R 4.1.2)
##  remotes               2.4.2   2021-11-30 [1] CRAN (R 4.1.2)
##  rlang                 0.4.12  2021-10-18 [1] CRAN (R 4.1.1)
##  rmarkdown           * 2.11    2021-09-14 [1] CRAN (R 4.1.1)
##  rprojroot             2.0.2   2020-11-15 [1] CRAN (R 4.1.0)
##  rstudioapi            0.13    2020-11-12 [1] CRAN (R 4.1.0)
##  rvest                 1.0.2   2021-10-16 [1] CRAN (R 4.1.1)
##  scales                1.1.1   2020-05-11 [1] CRAN (R 4.1.0)
##  sessioninfo           1.2.2   2021-12-06 [1] CRAN (R 4.1.2)
##  sjlabelled          * 1.1.8   2021-05-11 [1] CRAN (R 4.1.0)
##  SSACHR              * 1.0.0   2022-01-15 [1] Github (sjpierce/SSACHR@7ecfb11)
##  stringi               1.7.6   2021-11-29 [1] CRAN (R 4.1.2)
##  stringr               1.4.0   2019-02-10 [1] CRAN (R 4.1.0)
##  svglite               2.0.0   2021-02-20 [1] CRAN (R 4.1.0)
##  systemfonts           1.0.3   2021-10-13 [1] CRAN (R 4.1.1)
##  testthat              3.1.1   2021-12-03 [1] CRAN (R 4.1.2)
##  tibble                3.1.6   2021-11-07 [1] CRAN (R 4.1.2)
##  tidyr               * 1.1.4   2021-09-27 [1] CRAN (R 4.1.1)
##  tidyselect            1.1.1   2021-04-30 [1] CRAN (R 4.1.0)
##  tinytex               0.36    2021-12-19 [1] CRAN (R 4.1.2)
##  tmvnsim               1.0-2   2016-12-15 [1] CRAN (R 4.1.0)
##  tzdb                  0.2.0   2021-10-27 [1] CRAN (R 4.1.1)
##  usethis               2.1.5   2021-12-09 [1] CRAN (R 4.1.2)
##  utf8                  1.2.2   2021-07-24 [1] CRAN (R 4.1.0)
##  vctrs                 0.3.8   2021-04-29 [1] CRAN (R 4.1.0)
##  viridisLite           0.4.0   2021-04-13 [1] CRAN (R 4.1.0)
##  vistime             * 1.2.1   2021-04-10 [1] CRAN (R 4.1.0)
##  webshot               0.5.2   2019-11-22 [1] CRAN (R 4.1.0)
##  withr                 2.4.3   2021-11-30 [1] CRAN (R 4.1.2)
##  xfun                  0.29    2021-12-14 [1] CRAN (R 4.1.2)
##  xml2                  1.3.3   2021-11-30 [1] CRAN (R 4.1.2)
##  xtable                1.8-4   2019-04-21 [1] CRAN (R 4.1.0)
##  yaml                  2.2.1   2020-02-01 [1] CRAN (R 4.1.0)
##
##  [1] C:/Users/pierces1/OneDrive - Michigan State University/CSTATRedirects/Documents/R/win-library/4.1
##  [2] C:/Program Files/R/R-4.1.2/library
##
## ------------------------------------------------------------------------------
```

The current Git commit details and status are:

```
git_report()
```

```
## Local:    main S:/14-286/Analyses/SSACHR
## Remote:   main @ origin (https://github.com/sjpierce/SSACHR.git)
## Head:     [7ecfb11] 2022-01-15: Updated version number, date, and news.
##


## Untracked files:
##  Untracked:  inst/Step_01_Data_Mgt_Published_files/
```