Data_cleaning_BD

Sai

2025-07-06

Load the Necessary Libraries

```
library(tidyverse)
library(tidymodels)
library(skimr)
library(inspectdf)
library(janitor)
```

#Load the dataset.

```
crime <- read_csv("./data/crime_all.csv")
crime
```

```
## # A tibble: 4,980,995 x 26
##    `Incident Number` `Highest Offense Description` `Highest Offense Code`
##                <dbl> <chr>                                           <dbl>
##  1       20072790876 DISTURBANCE - OTHER                              3401
##  2       20065065520 FRAUD - OTHER                                    1199
##  3       20041101771 PROWLER                                          3414
##  4     2003421480455 CUSTODY ARREST TRAFFIC WARR                      3722
##  5       20052602038 DOC UNREASONABLE NOISE                           2405
##  6       20065065524 FRAUD - OTHER                                    1199
##  7       20135057728 PROTECTIVE ORDER                                 3829
##  8       20173300229 FAMILY DISTURBANCE                               3400
##  9       20035024083 THEFT FROM AUTO                                   603
## 10       20045056899 IDENTITY THEFT                                   4022
## # i 4,980,985 more rows
## # i 23 more variables: `Family Violence` <chr>, `Occurred Date Time` <dttm>,
## #   `Occurred Date Time - Year` <dbl>, `Occurred Date Time - Month` <dbl>,
## #   `Occurred Date Time - Week Of Year` <dbl>,
## #   `Occurred Date Time - Day` <dbl>, `Occurred Date Time - Day Of Week` <chr>,
## #   `Occurred Date Time - Hour` <dbl>, `Occurred Date Time - Minute` <dbl>,
## #   `Occurred Date Time - Seconds` <dbl>, `Occurred Date` <dttm>, ...
```

#Clean the names of all the columns for convenience with janitor package.

```
crime_cleaned <- clean_names(crime)
crime_cleaned
```

```
## # A tibble: 4,980,995 x 26
##    incident_number highest_offense_descri~1 highest_offense_code family_violence
##              <dbl> <chr>                                   <dbl> <chr>
##  1     20072790876 DISTURBANCE - OTHER                      3401 N
##  2     20065065520 FRAUD - OTHER                            1199 N
##  3     20041101771 PROWLER                                  3414 N
##  4   2003421480455 CUSTODY ARREST TRAFFIC ~                 3722 N
##  5     20052602038 DOC UNREASONABLE NOISE                   2405 N
##  6     20065065524 FRAUD - OTHER                            1199 N
##  7     20135057728 PROTECTIVE ORDER                         3829 N
##  8     20173300229 FAMILY DISTURBANCE                       3400 N
##  9     20035024083 THEFT FROM AUTO                           603 N
## 10     20045056899 IDENTITY THEFT                           4022 N
## # i 4,980,985 more rows
## # i abbreviated name: 1: highest_offense_description
## # i 22 more variables: occurred_date_time <dttm>,
## #   occurred_date_time_year <dbl>, occurred_date_time_month <dbl>,
## #   occurred_date_time_week_of_year <dbl>, occurred_date_time_day <dbl>,
## #   occurred_date_time_day_of_week <chr>, occurred_date_time_hour <dbl>,
## #   occurred_date_time_minute <dbl>, occurred_date_time_seconds <dbl>, ...
```

#Summarize the data distribution.

```
skim(crime_cleaned)
```

Table 1: Data summary

Name	crime_cleaned
Number of rows	4980995
Number of columns	26
Column type frequency:	
character	9
numeric	12
POSIXct	5
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
highest_offense_description	0	1.00	3	63	0	446	0
family_violence	0	1.00	1	1	0	3	0
occurred_date_time_day_of_week	203	1.00	6	9	0	7	0
location_type	35964	0.99	9	47	0	46	0
apd_sector	8857	1.00	1	6	0	66	0
apd_district	10257	1.00	1	5	0	74	0
clearance_status	1250571	0.75	1	1	0	4	0
ucr_category	3170557	0.36	3	3	0	16	0
category_description	3170557	0.36	4	18	0	7	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
incident_number	0	1.00	6.069281e+09	2.190937e+09	2.01035e+09	2.005298e+09	2.019e+09	2.021e+09	2.025251e+12	▇▁▁▁▁
highest_offense_code	0	1.00	1.700940e+03	1.221062e+03	103	601	1400	2716	8.905000e+03	▇▃▂▁▁
occurred_date_time_year	203	1.00	2.012320e+03	5.950000e+00	2002	2007	2012	2017	2.025000e+03	▇▆▆▆▂
occurred_date_time_month	203	1.00	6.480000e+00	3.410000e+00	1	4	6	9	1.200000e+01	▇▆▆▆▇
occurred_date_time_week_of_year	203	1.00	2.640000e+01	1.418900e+01	1	14	26	39	5.300000e+01	▆▆▆▇▆
occurred_date_time_day	203	1.00	1.560000e+01	8.860000e+00	1	8	16	23	3.100000e+01	▇▇▇▇▆
occurred_date_time_hour	203	1.00	1.260000e+01	6.064000e+00	0	7	13	18	2.300000e+01	▅▆▇▇▆
occurred_date_time_minute	203	1.00	2.230000e+01	1.889900e+01	0	0	21	38	5.900000e+01	▇▃▃▃▅
occurred_date_time_seconds	203	1.00	0.000000e+00	0.000000e+00	0	0	0	0	0.000000e+00	▁▁▇▁▁
occurred_time	62	1.00	1.318580e+03	7.139440e+02	0	800	1426	1930	2.400000e+03	▆▆▆▇▆
report_time	1	1.00	1.327110e+03	6.958650e+02	0	910	1406	1851	2.359000e+03	▆▆▆▇▆
council_district	43698	0.99	4.920000e+00	2.886000e+00	0	3	4	7	1.000000e+01	▇▇▆▃▂

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
occurred_date_time	297	1.00	2003-01-01 00:00:00	2025-04-06 04:55:00	2012-04-19 23:47:30	3091479
occurred_date	0	1.00	2003-01-01 00:00:00	2025-04-05 05:00:00	2012-04-19 05:00:00	15513
report_date_time	17	1.00	2002-11-29 05:30:00	2025-04-06 11:52:00	2012-04-28 16:23:30	3950586
report_date	0	1.00	2002-11-29 00:00:00	2025-04-06 05:00:00	2012-04-28 05:00:00	15517
clearance_date	714520	0.86	2003-01-01 00:00:00	2025-04-06 05:00:00	2012-09-12 05:00:00	15495

Correcting the Column in Mismatching Format.

The occurred_date, report_date columns are filled in POSIXct format on R. This column has date values in yyyy-mm-dd format. But a few cells of this column at the bottom have time as well in this format, for eg, 05:00:00. Removed the times on this column.

```
crime_cleaned <- crime_cleaned %>%
  mutate(occurred_date = as.Date(occurred_date))

crime_cleaned <- crime_cleaned %>%
  mutate(report_date = as.Date(report_date))
```

Found that a few rows at the bottom of the report_date_time (in POSIXct) column has values in this style "2016-03-18 22:16:00" by inspection. This column has the right dates but the times are wrong. There is another column report_time (in dbl) that has right times in hhmm format, but without colon (eg, 1910). Extracted the right times for the report_date_time column from report_time column. This was verified with the dataset (03-25) before merging.

Formatting Values.

By inspection, report_date_time column has a few rows at the bottom that has wrong times. The report_time has the right times, checked by inspection with (03-25) dataset. The report_date_time column has been formatted accordingly.

```
crime_cleaned[4980995, 16:18]
```

```
## # A tibble: 1 x 3
##   report_date_time    report_date report_time
##   <dttm>              <date>            <dbl>
## 1 2013-12-10 16:29:00 2013-12-10         1029
```

```
three_25 <- read_csv("./data/Crime_Reports_2(03-25).csv")
```

```
three_25[,8:10]
```

```
## # A tibble: 2,614,910 x 3
##    `Report Date Time` `Report Date` `Report Time`
##    <chr>              <chr>                <dbl>
##  1 11/29/2002  05:30  11/29/2002             530
##  2 01/01/2003  00:01  01/01/2003               1
##  3 01/01/2003  00:02  01/01/2003               2
##  4 01/01/2003  00:03  01/01/2003               3
##  5 01/01/2003  00:23  01/01/2003              23
##  6 01/01/2003  00:06  01/01/2003               6
##  7 01/01/2003  00:08  01/01/2003               8
##  8 01/01/2003  00:10  01/01/2003              10
##  9 01/01/2003  00:11  01/01/2003              11
## 10 01/01/2003  00:11  01/01/2003              11
## # i 2,614,900 more rows
```

```
rm(three_25)
```

```
library(lubridate)

crime_cleaned <- crime_cleaned %>%
  mutate(
    report_date_time = update(
      report_date_time,
      hour   = report_time %/% 100,      # integer division → HH
      minute = report_time %% 100,       # remainder → MM
      second = 0                         # reset seconds
    )
  )
```

```
crime_cleaned[4980995, 16:18]
```

```
## # A tibble: 1 x 3
##   report_date_time    report_date report_time
##   <dttm>              <date>            <dbl>
## 1 2013-12-10 10:29:00 2013-12-10         1029
```

Dealing with NAs for Temporal features.

I have found that occurred_date (POSIXct) column is full (in yyyy-mm-dd format, eg,2023-07-29) without any NAs. But the numeric columns, occurred_date_time_year, occurred_date_time_month, occurred_date_time_week_of_year, occurred_date_time_day, and the character column occurred_date_time_day_of_week have exactly 203 NAs. Extracted the dates from occurred date column and fill the columns which has NAs.

```
crime_cleaned %>%
  filter(
    if_any(
      c(occurred_date_time_year,
        occurred_date_time_month,
        occurred_date_time_week_of_year,
        occurred_date_time_day,
        occurred_date_time_day_of_week),
      is.na
    )
  ) %>%
  select(
    incident_number,
    occurred_date,
    occurred_date_time_year
  )
```

```
## # A tibble: 203 x 3
##    incident_number occurred_date occurred_date_time_year
##              <dbl> <date>                           <dbl>
##  1     20052602038 2005-09-17                          NA
##  2      2016160056 2016-01-16                          NA
##  3     20043360816 2004-12-01                          NA
##  4      2004440102 2004-02-13                          NA
##  5      2006831581 2006-03-24                          NA
##  6     20031900980 2003-07-09                          NA
##  7     20032141090 2003-08-02                          NA
##  8   2003924189381 2003-04-16                          NA
##  9   2003924939355 2003-02-06                          NA
## 10     20036001740 2003-08-29                          NA
## # i 193 more rows
```

```
crime_cleaned <- crime_cleaned %>%
  mutate(
    occurred_date_time_year         = coalesce(occurred_date_time_year, year(  occurred_date)),
    occurred_date_time_month        = coalesce(occurred_date_time_month, month( occurred_date)),
    occurred_date_time_week_of_year = coalesce(occurred_date_time_week_of_year, week(  occurred_date)),
    occurred_date_time_day          = coalesce(occurred_date_time_day, day(   occurred_date)),
    occurred_date_time_day_of_week = coalesce(
      occurred_date_time_day_of_week,
      weekdays(occurred_date))
  )
```

```
crime_cleaned %>%
  filter(
    if_any(
```

```
      c(occurred_date_time_year,
        occurred_date_time_month,
        occurred_date_time_week_of_year,
        occurred_date_time_day,
        occurred_date_time_day_of_week),
      is.na
    )
  )
```

```
## # A tibble: 0 x 26
## # i 26 variables: incident_number <dbl>, highest_offense_description <chr>,
## #   highest_offense_code <dbl>, family_violence <chr>,
## #   occurred_date_time <dttm>, occurred_date_time_year <dbl>,
## #   occurred_date_time_month <dbl>, occurred_date_time_week_of_year <dbl>,
## #   occurred_date_time_day <dbl>, occurred_date_time_day_of_week <chr>,
## #   occurred_date_time_hour <dbl>, occurred_date_time_minute <dbl>,
## #   occurred_date_time_seconds <dbl>, occurred_date <date>, ...
```

```
skim(crime_cleaned)
```

Table 5: Data summary

Name	crime_cleaned
Number of rows	4980995
Number of columns	26
Column type frequency:	
character	9
Date	2
numeric	12
POSIXct	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
highest_offense_description	0	1.00	3	63	0	446	0
family_violence	0	1.00	1	1	0	3	0
occurred_date_time_day_of_week	0	1.00	6	9	0	7	0
location_type	35964	0.99	9	47	0	46	0
apd_sector	8857	1.00	1	6	0	66	0
apd_district	10257	1.00	1	5	0	74	0
clearance_status	1250571	0.75	1	1	0	4	0
ucr_category	3170557	0.36	3	3	0	16	0
category_description	3170557	0.36	4	18	0	7	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
occurred_date	0	1	2003-01-01	2025-04-05	2012-04-19	8131
report_date	0	1	2002-11-29	2025-04-06	2012-04-28	8133

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
incident_number	0	1.00	6.069281e+12	2.190937e+11	2.0035	2.005298e+12	2.010503e+12	2.017114e+12	2.025251e+12	
highest_offense_code	0	1.00	1.700940e+03	1.231062e+03	103	601	1400	2716	8.905000e+03	
occurred_date_time_year	0	1.00	2.012320e+03	5.995000e+00	2002	2007	2012	2017	2.025000e+03	
occurred_date_time_month	0	1.00	6.480000e+00	3.461000e+00	1	4	6	9	1.200000e+01	
occurred_date_time_week_of_year	0	1.00	2.640000e+01	1.518900e+01	1	14	26	39	5.300000e+01	
occurred_date_time_day	0	1.00	1.560000e+01	8.816000e+00	1	8	16	23	3.100000e+01	
occurred_date_time_hour	203	1.00	1.260000e+01	6.614000e+00	0	7	13	18	2.300000e+01	
occurred_date_time_minute	203	1.00	2.230000e+01	1.889900e+01	0	0	21	38	5.900000e+01	
occurred_date_time_seconds	203	1.00	0.000000e+00	0.000000e+00	0	0	0	0	0.000000e+00	
occurred_time	62	1.00	1.318580e+03	7.139440e+02	2	800	1426	1930	2.400000e+03	
report_time	1	1.00	1.327110e+03	6.938650e+02	2	910	1406	1851	2.359000e+03	
council_district	43698	0.99	4.920000e+00	2.886000e+00	0	3	4	7	1.000000e+01	

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
occurred_date_time	297	1.00	2003-01-01 00:00:00	2025-04-06 04:55:00	2012-04-19 23:47:30	3091479
report_date_time	17	1.00	2002-11-29 05:30:00	2025-04-06 23:59:00	2012-04-28 17:28:00	2705091
clearance_date	714520	0.86	2003-01-01 00:00:00	2025-04-06 05:00:00	2012-09-12 05:00:00	15495

The “occurred_date_time” column (in POSIXct) with the format “2003-05-28 08:16:00”, has 297 NA rows. The “occurred_date” column (in Date) has no NAs. Extracted the dates and imputed the times as “00:00:00” for ““occurred_date_time”.

And imputed NA-filled cells of the numeric columns, namely, “occurred_date_time_hour”, “occurred_date_time_minute”, “occurred_date_time_seconds”, and “occurred_time” with “00:00:00”.

```
orig_tz <- attr(crime_cleaned$occurred_date_time, "tzone")[[1]]
na_idx <- is.na(crime_cleaned$occurred_date_time)

crime_cleaned$occurred_date_time[na_idx] <-
  as.POSIXct(
    paste(crime_cleaned$occurred_date[na_idx], "00:00:00"),
    tz = orig_tz
  )
```

```
crime_cleaned <- crime_cleaned %>%
  mutate(
    # pull out components from your now-complete POSIXct
    h = hour(  occurred_date_time),
```

```
    m = minute(occurred_date_time),
    s = second(occurred_date_time),

    # fill the numeric columns only where they're still NA
    occurred_date_time_hour    = coalesce(occurred_date_time_hour,    h),
    occurred_date_time_minute  = coalesce(occurred_date_time_minute,  m),
    occurred_date_time_seconds = coalesce(occurred_date_time_seconds, s),

    # fill your "seconds since midnight" field
    occurred_time = coalesce(occurred_time, h * 3600 + m * 60 + s)
  ) %>%
  select(-h, -m, -s)
```

```
sum(is.na(crime_cleaned$occurred_date_time))
```

```
## [1] 0
```

```
sum(is.na(crime_cleaned$occurred_date_time_hour))
```

```
## [1] 0
```

```
sum(is.na(crime_cleaned$occurred_date_time_minute))
```

```
## [1] 0
```

```
sum(is.na(crime_cleaned$occurred_date_time_seconds))
```

```
## [1] 0
```

```
sum(is.na(crime_cleaned$occurred_time))
```

```
## [1] 0
```

Retrieving the right values.

Now "occurred_date_time" column is perfect. I see that a few things on the correct column "occurred_date_time" is not matching with occurred_date_time_year, occurred_date_time_month, occurred_date_time_hour, occurred_date_time_week_of_year, occurred_date_time_day, occurred_date_time_day_of_w occurred_date_time_hour, occurred_date_time_minute, occurred_date_time_seconds. Extracted the right dates for these columns from the correct column ""occurred_date_time".

occurred_date_time_day_of_week is also converted from character to factor.

```
crime_cleanedi <- crime_cleaned %>%
  mutate(
    occurred_date_time_year         = year(    occurred_date_time),
    occurred_date_time_month        = month(   occurred_date_time),
    occurred_date_time_week_of_year = isoweek( occurred_date_time),
```

```
    occurred_date_time_day          = day(     occurred_date_time),
    occurred_date_time_day_of_week  = wday(    occurred_date_time,
                                           label = TRUE,
                                           abbr  = FALSE),
    occurred_date_time_hour         = hour(    occurred_date_time),
    occurred_date_time_minute       = minute(  occurred_date_time),
    occurred_date_time_seconds      = second(  occurred_date_time)
  )
```

```
rm(crime) #Removed for convenience
```

```
crime_cleaned <- crime_cleanedi #Created for convenience
```

```
skim(crime_cleaned)
```

Table 10: Data summary

Name	crime_cleaned
Number of rows	4980995
Number of columns	26
Column type frequency:	
character	8
Date	2
factor	1
numeric	12
POSIXct	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
highest_offense_description	0	1.00	3	63	0	446	0
family_violence	0	1.00	1	1	0	3	0
location_type	35964	0.99	9	47	0	46	0
apd_sector	8857	1.00	1	6	0	66	0
apd_district	10257	1.00	1	5	0	74	0
clearance_status	1250571	0.75	1	1	0	4	0
ucr_category	3170557	0.36	3	3	0	16	0
category_description	3170557	0.36	4	18	0	7	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
occurred_date	0	1	2003-01-01	2025-04-05	2012-04-19	8131
report_date	0	1	2002-11-29	2025-04-06	2012-04-28	8133

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
occurred_date_time_day_of_week	0	1	TRUE	7	Fri: 750712, Sat: 745649, Sun: 706118, Thu: 698127

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
incident_number	0	1.00	6.069281e+12	2.100937e+12	2.0035	2.005298e+12	2.010503e+12	2.017114e+12	2.025251e+12	
highest_offense_code	0	1.00	1.700940e+03	1.221062e+03	103	601	1400	2716	8.905000e+03	
occurred_date_time_year	0	1.00	2.012320e+03	5.950000e+00	2003	2007	2012	2017	2.025000e+03	
occurred_date_time_month	0	1.00	6.470000e+00	3.410000e+00	1	4	6	9	1.200000e+01	
occurred_date_time_week_of_year	0	1.00	2.640000e+01	1.518900e+01	1	14	26	39	5.300000e+01	
occurred_date_time_day	0	1.00	1.556000e+01	8.816000e+00	1	8	16	23	3.100000e+01	
occurred_date_time_hour	0	1.00	1.198000e+01	7.417000e+00	0	5	13	19	2.300000e+01	
occurred_date_time_minute	0	1.00	2.230000e+01	1.839900e+01	0	0	21	38	5.900000e+01	
occurred_date_time_seconds	0	1.00	0.000000e+00	0.000000e+00	0	0	0	0	0.000000e+00	
occurred_time	0	1.00	1.318560e+03	7.139450e+02	2	800	1426	1930	2.400000e+03	
report_time	1	1.00	1.327110e+03	6.958650e+02	2	910	1406	1851	2.359000e+03	
council_district	43698	0.99	4.920000e+00	2.886000e+00	0	3	4	7	1.000000e+01	

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
occurred_date_time	0	1.00	2003-01-01 00:00:00	2025-04-06 04:55:00	2012-04-19 20:51:00	3091481
report_date_time	17	1.00	2002-11-29 05:30:00	2025-04-06 23:59:00	2012-04-28 17:28:00	2705091
clearance_date	714520	0.86	2003-01-01 00:00:00	2025-04-06 05:00:00	2012-09-12 05:00:00	15495

Removal of Redundant Features & Cells.

To crime_cleaned dataset performed the following things;-

- Removed incident_number, ucr_category, category_description, and occurred_date_time_seconds columns fully.
- Remove the cells that has NAs in location_type, apd_sector, apd_district, clearance_status, clearance_date, report_date_time.

```
# Step 1: Remove the unwanted columns
crime_cleaned <- crime_cleaned %>%
  select(-incident_number, -ucr_category, -category_description, -occurred_date_time_seconds)

# Step 2: Drop any rows where any of these six columns is NA
crime_cleaned <- crime_cleaned %>%
  filter(
    !is.na(location_type),
```

```
    !is.na(apd_sector),
    !is.na(apd_district),
    !is.na(clearance_status),
    !is.na(clearance_date),
    !is.na(report_date_time)
  )
```

```
crime_cleaned <- crime_cleaned %>%
  select(-council_district)
```

```
skim(crime_cleaned)
```

Table 16: Data summary

Name	crime_cleaned
Number of rows	3646608
Number of columns	21
Column type frequency:	
character	6
Date	2
factor	1
numeric	9
POSIXct	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
highest_offense_description	0	1	3	63	0	442	0
family_violence	0	1	1	1	0	3	0
location_type	0	1	9	47	0	46	0
apd_sector	0	1	1	5	0	40	0
apd_district	0	1	1	5	0	39	0
clearance_status	0	1	1	1	0	4	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
occurred_date	0	1	2003-01-01	2025-04-05	2013-12-25	8131
report_date	0	1	2003-01-01	2025-04-05	2014-01-05	8131

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
occurred_date_time_day_of_week	0	1	TRUE	7	Fri: 549299, Sat: 547103, Sun: 520121, Mon: 510738

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
highest_offense_code	0	1	1662.68	1199.44	100	601	1199	2703	8905	
occurred_date_time_year	0	1	2013.66	5.53	2003	2009	2013	2018	2025	
occurred_date_time_month	0	1	6.48	3.43	1	4	6	9	12	
occurred_date_time_week_of_year	0	1	26.42	14.97	1	13	26	39	53	
occurred_date_time_day	0	1	15.56	8.86	1	8	16	23	31	
occurred_date_time_hour	0	1	11.94	7.49	0	5	13	19	23	
occurred_date_time_minute	0	1	22.75	18.98	0	1	22	39	59	
occurred_time	0	1	1311.70	723.33	0	800	1424	1928	2400	
report_time	0	1	1329.49	664.00	0	914	1411	1855	2359	

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
occurred_date_time	0	1	2003-01-01 00:00:00	2025-04-06 04:00:00	2013-12-25 20:12:30	2467152
report_date_time	0	1	2003-01-01 00:03:00	2025-04-06 23:00:00	2014-01-05 14:45:00	2073565
clearance_date	0	1	2003-01-01 00:00:00	2025-04-06 05:00:00	2014-01-29 00:00:00	15491

Type-setting a few columns.

In order to reduce redundancy and discrepancy of multiple data types, two columns (in date format) were converted to POSIXct format.

```
crime_cleanedi <- crime_cleaned %>%
  mutate(
    occurred_date = as.POSIXct(occurred_date),
    report_date   = as.POSIXct(report_date)
  )
```

```
crime_cleaned <- crime_cleanedi
```

Changing to the right formats.

By inspection, found that a few rows at the bottom of the feature clearance_date contained times, for eg 17:00:00. With the conversion of the whole column to date format, the issue was addressed. And then the column was again converted to POSIXct due to the reasons stated previously.

```r
crime_cleaned <- crime_cleaned %>% mutate(
  clearance_date = as_date(clearance_date)
)

crime_cleaned <- crime_cleaned %>% mutate(
  clearance_date = as.POSIXct(clearance_date)
)
```

Addressal and Removal of Variable categories.

The family_violence column, which is supposed to have values in either Y/N had a few values of "n". occurred_date_time_year, occurred_date_time_month were correctly present. The occurred_date_time_week_of_year column has been checked for the years with 53 weeks and found that in the years 2005, 2010, 2015, and 2021 had only 52 weeks with calendar. The apd_sector and apd_district had values that were related to the original values and had to be unified with the same names. For eg, BAKR of apd_sector has been renamed to BA, etc.

```r
count(crime_cleaned, family_violence)
```

```
## # A tibble: 3 x 2
##   family_violence       n
##   <chr>             <int>
## 1 N               3353256
## 2 Y                293106
## 3 n                   246
```

```r
years <- count(crime_cleaned, occurred_date_time_year)
rm(years)

months <- count(crime_cleaned, occurred_date_time_month)
rm(months)

woy <- count(crime_cleaned, occurred_date_time_week_of_year)
rm(months)

woy_53 <- crime_cleaned %>% filter(
  occurred_date_time_week_of_year == 53
)

woy_53y <- count(woy_53, occurred_date_time_year)

rm(woy, woy_53, woy_53y)

day <- count(crime_cleaned, occurred_date_time_day)
rm(day)

dow <- count(crime_cleaned, occurred_date_time_day_of_week)
rm(dow)

h <- count(crime_cleaned, occurred_date_time_hour)
m <- count(crime_cleaned, occurred_date_time_minute)
```

```r
rm(h,m)

apds <- count(crime_cleaned, apd_sector)

apdd <- count(crime_cleaned, apd_district)
```

Renaming the values of apd_sector and apd_district columns.

On the apd_sector (in character) column, removed the cells with values that are 2, 8, 83, 88, 99, A, A1, AS, AV, F6, G, RD, UT.

In the same column, changed the variables that are in different names to a single unifying name. Converted AD, ADAM to AD; Converted BA, BAKER, BAKR to BA; Converted CH, C, CHAR to CH; Converted DA, D, DAVD, DAVID to DA; Converted ED, E, EDWD to ED; Converted FR, FRNK, FRK to FR; Converted G, GE to GE; Converted HE, HENR, HENRY, to HE; Converted I, ID, IDA to ID;

Verified that everything is in the right order by inspection.

```r
crime_cleaned <- crime_cleaned %>%
  # 1) remove rows whose apd_sector is in the "drop" list
  filter(
    !apd_sector %in% c(
      "2","8","83","88","99",
      "A","A1","AS","AV","F6",
      "G","RD","UT"
    )
  ) %>%
  # 2) collapse all variants down to your canonical codes
  mutate(
    apd_sector = case_when(
      apd_sector %in% c("AD",   "ADAM")           ~ "AD",
      apd_sector %in% c("BA",   "BAKER", "BAKR")  ~ "BA",
      apd_sector %in% c("CH",   "C",     "CHAR")  ~ "CH",
      apd_sector %in% c("DA",   "D",     "DAVD","DAVID") ~ "DA",
      apd_sector %in% c("ED",   "E",     "EDWD") ~ "ED",
      apd_sector %in% c("FR",   "FRNK",  "FRK")  ~ "FR",
      apd_sector %in% c("GE")                     ~ "GE",
      apd_sector %in% c("HE",   "HENR",  "HENRY", "HR")~ "HE",
      apd_sector %in% c("I",    "ID",    "IDA")  ~ "ID",
      TRUE                                        ~ apd_sector
    )
  )

apds_cleaned <- count(crime_cleaned, apd_sector)

apds
```

```
## # A tibble: 40 x 2
##    apd_sector      n
##    <chr>       <int>
##  1 2               2
##  2 8               2
```

```
##  3 83              2
##  4 88           8765
##  5 99              2
##  6 A               4
##  7 A1              2
##  8 AD         358459
##  9 ADAM            3
## 10 AP           21462
## # i 30 more rows
```

```
apds_cleaned
```

```
## # A tibble: 10 x 2
##    apd_sector      n
##    <chr>       <int>
##  1 AD         358462
##  2 AP          21462
##  3 BA         377641
##  4 CH         446323
##  5 DA         454824
##  6 ED         490365
##  7 FR         437482
##  8 GE         248944
##  9 HE         411492
## 10 ID         388982
```

```
rm(apds, apds_cleaned)
```

On the apd_district (in character) column, removed the cells with values that are 0, 9, 99, A, C, D, D10, D9, DAVID, P, S.

On the apd_district (in character) column, changed the variables that are in different names to a single unifying name. Converted 1, 10, 11, 12, 01, I1 to 1; Converted 2, A2, D2 to 2; Converted 4, 493, 04, A4, D4, I4 to 4; Converted 7, B7, C7, D7 to 7; Converted 8, 83, 88, C8 to 8.

```
crime_cleaned <- crime_cleaned %>%
  # 1) remove rows whose apd_district is in the "drop" list
  filter(
    !apd_district %in% c(
      "0","9","99","A","C","D","D10","D9","DAVID","P","S"
    )
  ) %>%
  # 2) unify all remaining values to your five canonical districts
  mutate(
    apd_district = case_when(
      apd_district %in% c("1",  "10", "11", "12", "01", "I1") ~ "1",
      apd_district %in% c("2",  "A2", "D2")                   ~ "2",
      apd_district %in% c("4",  "493","04", "A4", "D4", "I4")~ "4",
      apd_district %in% c("7",  "B7", "C7", "D7")             ~ "7",
      apd_district %in% c("8",  "83", "88", "C8")             ~ "8",
      TRUE                                                     ~ apd_district
    )
  )
```

```
apdd_cleaned <- count(crime_cleaned, apd_district)

apdd
```

```
## # A tibble: 39 x 2
##    apd_district      n
##    <chr>         <int>
##  1 0               218
##  2 01                2
##  3 04                2
##  4 1            726507
##  5 10              553
##  6 11               39
##  7 12               37
##  8 2            763453
##  9 3            463866
## 10 4            427550
## # i 29 more rows
```

```
apdd_cleaned
```

```
## # A tibble: 8 x 2
##   apd_district      n
##   <chr>         <int>
## 1 1            727142
## 2 2            763457
## 3 3            463866
## 4 4            427558
## 5 5            361648
## 6 6            300589
## 7 7            329510
## 8 8            236605
```

```
rm(apdd_cleaned, apdd)
```

#Dropping a type of the outcome variable.

In order to adhere to the primary research question, where our goal was to predict the probability of a case solved status as "C", I have converted the category "O"to "C", where "O" represents that a case is cleared by arrest or other means, and "C" represents that a case has been solved. And dropped the cells which had a wrong type called "9" on the clearance_status column.

```
count(crime_cleaned, clearance_status)
```

```
## # A tibble: 4 x 2
##   clearance_status       n
##   <chr>              <int>
## 1 9                      2
## 2 C                1140788
## 3 N                2308102
## 4 O                 161483
```

```
crime_cleaned <- crime_cleaned %>%
  mutate(
    clearance_status = if_else(clearance_status == "O", "C", clearance_status)
  )
crime_cleaned <- crime_cleaned %>%
  filter(clearance_status != "9")

count(crime_cleaned, clearance_status)
```

```
## # A tibble: 2 x 2
##   clearance_status       n
##   <chr>              <int>
## 1 C                1302271
## 2 N                2308102
```

Cleaning other variables.

The lower case "n" of family_violence which is a wrong type acccording to majority classes, has been changed to "N". The values that was on the 53rd week on the years 2005, 2010, 2015, and 2021 were dropped, as they were inspected to such values.

```
count(crime_cleaned, family_violence)
```

```
## # A tibble: 3 x 2
##   family_violence       n
##   <chr>             <int>
## 1 N               3318331
## 2 Y                291796
## 3 n                   246
```

```
crime_cleaned <- crime_cleaned %>%
  # 1) Fix any lowercase "n" to "N" (also handles any lowercase "y" if you prefer)
  mutate(
    family_violence = toupper(family_violence)
  ) %>%
  # 2) Drop ISO-week 53 in the years that never have 53 weeks
  filter(
    !(occurred_date_time_week_of_year == 53 &
      occurred_date_time_year %in% c(2005, 2010, 2015, 2021))
  )

count(crime_cleaned, family_violence)
```

```
## # A tibble: 2 x 2
##   family_violence       n
##   <chr>             <int>
## 1 N               3312729
## 2 Y                291265
```

```r
woy_53 <- crime_cleaned %>% filter(
  occurred_date_time_week_of_year == 53
)

woy_53y <- count(woy_53, occurred_date_time_year)

woy_53y
```

```
## # A tibble: 4 x 2
##   occurred_date_time_year     n
##                     <dbl> <int>
## 1                    2004  1313
## 2                    2009  2415
## 3                    2016  1906
## 4                    2020  1765
```

```r
rm(woy_53, woy_53y)
```

```r
skim(crime_cleaned)
```

Table 22: Data summary

Name	crime_cleaned
Number of rows	3603994
Number of columns	21
Column type frequency:	
character	6
factor	1
numeric	9
POSIXct	5
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
highest_offense_description	0	1	3	63	0	438	0
family_violence	0	1	1	1	0	2	0
location_type	0	1	9	47	0	46	0
apd_sector	0	1	2	2	0	10	0
apd_district	0	1	1	1	0	8	0
clearance_status	0	1	1	1	0	2	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
occurred_date_time_day_of_week	0	1	TRUE	7	Fri: 542287, Sat: 540813, Sun: 513936, Mon: 504908

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
highest_offense_code	0	1	1657.38	1196.95	100	601	1199	2703	8905	
occurred_date_time_year	0	1	2013.67	5.53	2003	2009	2013	2018	2025	
occurred_date_time_month	0	1	6.48	3.42	1	4	6	9	12	
occurred_date_time_week_of_year	0	1	26.37	14.94	1	13	26	39	53	
occurred_date_time_day	0	1	15.56	8.85	1	8	16	23	31	
occurred_date_time_hour	0	1	11.93	7.49	0	5	13	19	23	
occurred_date_time_minute	0	1	22.73	18.98	0	1	22	39	59	
occurred_time	0	1	1312.27	724.39	0	800	1426	1930	2400	
report_time	0	1	1329.75	664.91	0	913	1412	1856	2359	

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
occurred_date_time	0	1	2003-01-01 00:00:00	2025-04-06 04:00:00	2013-12-27 00:07:00	2440600
occurred_date	0	1	2003-01-01 00:00:00	2025-04-05 00:00:00	2013-12-26 00:00:00	8123
report_date_time	0	1	2003-01-01 00:03:00	2025-04-06 23:00:00	2014-01-06 11:08:00	2052642
report_date	0	1	2003-01-01 00:00:00	2025-04-05 00:00:00	2014-01-06 00:00:00	8131
clearance_date	0	1	2003-01-01 00:00:00	2025-04-06 00:00:00	2014-01-29 00:00:00	8120

#Checked if the location types column is fine.

```
lt <- count(crime_cleaned, location_type)
lt
```

```
## # A tibble: 46 x 2
##    location_type                              n
##    <chr>                                  <int>
##  1 ABANDONED/CONDEMNED STRUCTURE           2278
##  2 AIR / BUS / TRAIN TERMINAL              9964
##  3 AMUSEMENT PARK                           245
##  4 ARENA / STADIUM / FAIRGROUNDS / COLISEUM  540
##  5 ATM SEPARATE FROM BANK                   697
##  6 AUTO DEALERSHIP NEW / USED              4073
##  7 BANK / SAVINGS & LOAN                  13986
##  8 BAR / NIGHTCLUB                        41564
##  9 CAMP / CAMPGROUND                        818
## 10 CHURCH / SYNAGOGUE / TEMPLE / MOSQUE    8665
## # i 36 more rows
```

```
rm(lt)
```

#Typesetting each column.

The types of data are four according to conventions. They are-

Quantitative variables are numbers that we can measure.

a) Quantitative Discrete - The measurable numbers , in smallest units, which can not be infinitely divided. The format must be integers.

b) Quantitative Continuous - The measurable numbers that can be infinitely divided into smaller units, like time, length, weight, or temperature, and anything that is derived from them. The format must be numeric.

Categorical variables are things that can be classified with labels.

c) Categorical Nominal - The labels that are only given for identification or describing the nature, and they do not have an inherent order. The format must be factor (nature) or character(id).

d) Categorical Ordinal - Categorical ordinal are labels that have an order — for example, the bronze, silver and gold medals in the Olympics. The format must be ordered.

To look at how the columns of the data look like and the type they are in, every column has been discussed below.

i) Incident Number is a character. This is correct type according to conventions.

ii) Highest Offense Description is a character. This is correct type according to conventions. The correct type is factor as it describes the nature of incidents.

The following features were converted to the right type setting format of R.

```
crime_cleaned <- crime_cleaned %>%
  # 1) Convert these character columns to factors
  mutate(
    across(
      c(
        highest_offense_description,
        family_violence,
        location_type,
        apd_sector,
        apd_district,
        clearance_status
      ),
      factor
    )
  ) %>%
  # 2) Convert all date-time components to the respective types:
  mutate(
    # year as integer
    occurred_date_time_year         = as.integer(occurred_date_time_year),
    # month number → ordered factor Jan-Dec
```

```
    occurred_date_time_month        = factor(
                                        month.abb[occurred_date_time_month],
                                        levels  = month.abb,
                                        ordered = TRUE
                                      ),
    # week / day / hour / minute / second as integers
    occurred_date_time_week_of_year = as.integer(occurred_date_time_week_of_year),
    occurred_date_time_day          = as.integer(occurred_date_time_day),
    occurred_date_time_hour         = as.integer(occurred_date_time_hour),
    occurred_date_time_minute       = as.integer(occurred_date_time_minute),
    # day_of_week → ordered factor Monday-Sunday
    occurred_date_time_day_of_week  = factor(
                                        occurred_date_time_day_of_week,
                                        levels  = c(
                                          "Monday","Tuesday","Wednesday",
                                          "Thursday","Friday","Saturday","Sunday"
                                        ),
                                        ordered = TRUE
                                      )
  )

crime_cleaned <- crime_cleaned %>% mutate(
  highest_offense_code = as.character(highest_offense_code)
)
```

```
skim(crime_cleaned)
```

Table 27: Data summary

Name	crime_cleaned
Number of rows	3603994
Number of columns	21
Column type frequency:	
character	1
factor	8
numeric	7
POSIXct	5
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
highest_offense_code	0	1	3	4	0	396	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
highest_offense_description	0	1	FALSE	438	BUR: 322920, THE: 322427, FAM: 240847, CRI: 184294
family_violence	0	1	FALSE	2	N: 3312729, Y: 291265
occurred_date_time_month	0	1	TRUE	12	Mar: 314135, May: 311315, Jul: 310873, Aug: 309566
occurred_date_time_day_of_week	0	1	TRUE	7	Fri: 542287, Sat: 540813, Sun: 513936, Mon: 504908
location_type	0	1	FALSE	46	RES: 1394250, HWY: 824225, PAR: 425204, COM: 207614
apd_sector	0	1	FALSE	10	ED: 489492, DA: 450425, CH: 445207, FR: 436608
apd_district	0	1	FALSE	8	2: 762112, 1: 725998, 3: 462978, 4: 426779
clearance_status	0	1	FALSE	2	N: 2303668, C: 1300326

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
occurred_date_time_year	0	1	2013.67	5.53	2003	2009	2013	2018	2025	
occurred_date_time_week_of_year	0	1	26.37	14.94	1	13	26	39	53	
occurred_date_time_day	0	1	15.56	8.85	1	8	16	23	31	
occurred_date_time_hour	0	1	11.93	7.49	0	5	13	19	23	
occurred_date_time_minute	0	1	22.73	18.98	0	1	22	39	59	
occurred_time	0	1	1312.27	724.39	0	800	1426	1930	2400	
report_time	0	1	1329.75	664.91	0	913	1412	1856	2359	

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
occurred_date_time	0	1	2003-01-01 00:00:00	2025-04-06 04:00:00	2013-12-27 00:07:00	2440600
occurred_date	0	1	2003-01-01 00:00:00	2025-04-05 00:00:00	2013-12-26 00:00:00	8123
report_date_time	0	1	2003-01-01 00:03:00	2025-04-06 23:00:00	2014-01-06 11:08:00	2052642
report_date	0	1	2003-01-01 00:00:00	2025-04-05 00:00:00	2014-01-06 00:00:00	8131
clearance_date	0	1	2003-01-01 00:00:00	2025-04-06 00:00:00	2014-01-29 00:00:00	8120

Extract the test data of 2025.

```
library(readr)
crime_test <- crime_cleaned %>%
  filter(occurred_date_time_year == 2025)
```

```r
# 2. Save crime_test to a CSV file
# write_csv(crime_test, "crime_test.csv") #For Saving file
```

#Obtain the Main Data.

```r
crime_main <- crime_cleaned %>%
  filter(occurred_date_time_year != 2025)

# write_csv(crime_main, "crime_main.csv") #For Saving file
```

Sampling

Due to restriction of the processing capabilities of the computers, An average of 0.92 million rows were sampled. The seed was my student numer 1906525.

```r
# 1. Stratified sampling ------------------------------------------------

set.seed(1906525)

# compute the overall fraction needed
frac <- 1.25e6 / nrow(crime_main)

crime_sample <- crime_main %>%
  group_by(
    occurred_date_time_year,
    occurred_date_time_month,
    occurred_date_time_day,
    location_type
  ) %>%
  slice_sample(prop = frac) %>%
  ungroup()

# make sure we got roughly 1 M
nrow(crime_sample)
```

```
## [1] 1174559
```

```r
# write_csv(crime_sample, "crime_sample.csv") #For Saving file
```