Data Cleaning Process

```
library(here)
library(readr)
library(dplyr)
library(tidyr)
library(purrr)
library(ggplot2)
library(magrittr)
library(lubridate)
library(data.table)
```

Officers

- **birth_year**: Birth year of the officer.
- **appointed_month**: Month officer was appointed.
- **officer_id**: Unique ID for each officer. **Unique identifier.**
- **officer_race**: Race of the officer.
- **officer_gender**: Gender of the officer.
- **spanish**: Does the officer speak Spanish or not?
- Uniquely identified by **officer_id**.

Shift Assignments

- **officer_id**: Unique ID for each officer.
- **month**: Month of the shift in year-month-date format.
- **rank**: Rank of the officer assigned to the shift.
- **unit**: Unit of the officer assigned to the shift.
- **date**: Date of the shift.
- **shift**: The shift the officer is assigned to.
- **start_time**: Hour start time of the shift in military time.
- **end_time**: Hour end of the shift in military time.
- **weekday**: Day of the week of the shift.
- **beat_assigned**: The beat the officer is assigned to.
- **appointed_month**: Month the officer was appointed.
- **months_from_start**: Months between officer appointment and shift date.
- **months_from_start_sq**: Months between officer appointment and shift date squared.
- **duration**: Length of the shift in hours.
- Uniquely identified by **officer_id** and **date**.

```
officers <- read_csv(here("bocar_data", "officers.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   birth_year = col_double(),
##   appointed_month = col_date(format = ""),
##   officer_id = col_double(),
##   officer_race = col_character(),
```

```
##   officer_gender = col_character(),
##   spanish = col_double()
## )
```

```
assignments <- read_csv(here("bocar_data", "assignments.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   officer_id = col_double(),
##   month = col_date(format = ""),
##   rank = col_character(),
##   unit = col_double(),
##   date = col_date(format = ""),
##   shift = col_double(),
##   start_time = col_double(),
##   end_time = col_double(),
##   weekday = col_character(),
##   beat_assigned = col_character(),
##   appointed_month = col_date(format = ""),
##   months_from_start = col_double(),
##   months_from_start_sq = col_double(),
##   duration = col_double()
## )
```

Number of Officers: 33645
Number of Shift Assignments: 3519518

Data Checks

- Check understanding to make sure columns mean what I think they mean.
- Drop the month and months_from_start_sq columns from the shift assignment data. They can be recreated as needed. Also drop appointed_month since that is redundant with the officer data.

```
check <-
    assignments %>%
    select(date, month, weekday, appointed_month, months_from_start) %>%
    mutate(month_check = floor_date(date, unit = "month"),
           weekday_check = wday(date),
           months_f_start_check = interval(appointed_month, date) %/% months(1),
           weekday_check = case_when(weekday_check == 1 ~ "Sun",
                                     weekday_check == 2 ~ "Mon",
                                     weekday_check == 3 ~ "Tue",
                                     weekday_check == 4 ~ "Wed",
                                     weekday_check == 5 ~ "Thu",
                                     weekday_check == 6 ~ "Fri",
                                     weekday_check == 7 ~ "Sat"))

assignments <-
    assignments %>%
    select(-month, -appointed_month,-months_from_start_sq)
```

Month Check: 3519518
Weekday Check: 3519518
Months From Start Check: 3518740

Summary Statistics

```
GetNumericSummary <- function(df, col) {

    summary <- summary(df[[col]])
    if("NAs" %in% names(attributes(summary)))
        names <- c(names(summary), "NA's")
    else
        names <- names(summary)

    tibble(values = as.character(summary), variable = names) %>%
        pivot_wider(names_from = variable, values_from = values) %>%
        mutate(variable = col)
}

GetCategoricalSummary <- function(df, col) {

    df %>%
        group_by(.data[[col]]) %>%
        summarise(n = n()) %>%
        ungroup() %>%
        mutate(prcnt = n / sum(n)) %>%
        pivot_longer(col, values_transform = list(value = as.character))
}

numericSummaryOfficers <-
    map_dfr(c("birth_year", "appointed_month"),
            GetNumericSummary,
            df = officers)

categoricalSummaryOfficers <-
    map_dfr(c("officer_race", "officer_gender", "spanish"),
            GetCategoricalSummary,
            df = officers)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(col)` instead of `col` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
numericSummaryAssignments <-
    map_dfr(c("date", "start_time", "end_time", "months_from_start", "duration"),
            GetNumericSummary,
            df = assignments)

categoricalSummaryAssignments <-
    map_dfr(c("rank", "shift", "unit", "weekday"),
            GetCategoricalSummary,
            df = assignments)
```

Number of unique beats: 6616
Number of non-missing beat assignments: 3519518
Percentage of non-missing beat assignments: 1

Number of shifts which are between 8 and 9 hours: 3101343

Percentage of shifts which are between 8 and 9 hours: 0.881184

Data Filtering

- Only keep shift assignments for those with a rank of police officer.
- Only keep Black, White, and Hispanic officers.

```
assignmentsFltr <- assignments %>% filter(rank == "POLICE OFFICER")

officersFltr <-
    officers %>%
    filter(officer_race %in%
               c("officer_black", "officer_white", "officer_hisp"))
```

Number of Officers: 32887
Number of Shift Assignments: 3050853

Join officers to their shift assignments

```
# Join officers to their assignments
officerAssignment <-
    assignmentsFltr %>%
    inner_join(officersFltr, by = "officer_id")
write_csv(officerAssignment, here("bocar_data", "officerAssignment.csv"))

# Anti join assignments
antiJoinAssignment <-
    anti_join(assignmentsFltr, officersFltr, by = "officer_id")

assignmentCheck <-
    filter(officers, officer_id %in% antiJoinAssignment$officer_id)

table(assignmentCheck$officer_race)
```

```
##
##   officer_aapi officer_native
##            241             26
```

```
# Anti join officers
antiJoinOfficer <- anti_join(officersFltr, assignmentsFltr, by = "officer_id")
rankCheck <- filter(assignments, officer_id %in% antiJoinOfficer$officer_id)
length(unique(rankCheck$officer_id)) / nrow(antiJoinOfficer)
```

```
## [1] 0.04841943
```

- Number of officer assignments: 2932680
- Rows can be uniquely identified using **officer_id** and **date**.
- Number of non-matching assignments: 118173
- Percentage of assignments which didn't match: 0.0387344
  - If an assignment doesn't match, it's because the assignment was for an officer who isn't Black, Hispanic, or White.
- Number of non-matching officers: 25940
- Percentage of non-matching officers: 0.7887615

  - Number of officers who didn't match because of their rank: 1256
  - Percentage of non-matching officers due to rank: 0.0484194

- Number matching officers: 6947
- Percentage of matching officers: 0.2112385
  - If an officer doesn't match, it's either because of the officer's rank **OR** because the officer didn't have any assignments during this time period.
  - The question for us is: Why are there are officers who didn't have any assignments during this time period? The most obvious explanation is they retired/quit. However it is worth investigating if there are other reasons which could call into question the validity of our findings.