OSEI Senior Data Analyst Exercise: Exploratory Analysis

Mayra Smith-Coronado

2024-06-24

Intro

The included datasets come from two tables within the DSD relational database: Incarceration and Person. The incarceration table's unit of analysis is one booking into the jail and includes booking-related data such as the times into and out of the jail. The person table's unit of analysis is one person and includes demographic information like age and race. The "Person_id" column is the common variable between them. (For this exercise, the real Person_id has been suppressed and replaced with a unique random number to protect identities.)

The point of this exercise is to demonstrate how you think analytically as much as it is to arrive at the "correct" answers. Please provide your best answers to the questions below, using the tools and methods you deem most effective. Please submit written answers in a clear and concise form by the deadline. **Please also share your code so we can review it.**

```
# Load in Incarceration Data ------------------------------------------------
# note time is not going to be read in for this data set, only date information
incarceration_data <- read.xlsx("~/osei_data_exercise/01 - data/Incarceration.xlsx") %>%
  # convert to tibble
  tibble() %>%
  # update column names to follow snake case naming convention
  janitor::clean_names() %>%
  # correctly read excel dates as dates instead of numeric values
  mutate(across(.cols = c("date_in", "release_out"),
                .fns = janitor::excel_numeric_to_date))

# check to make sure that there are no duplicate booking numbers and that
# each row represents one booking
incarceration_data %>%
  count(booking_number) %>%
  filter(n > 1)
```

```
## # A tibble: 0 x 2
## # i 2 variables: booking_number <chr>, n <int>
```

```
# Load in Person Data -------------------------------------------------------
person_data <- read.xlsx("~/osei_data_exercise/01 - data/Person.xlsx") %>%
  # convert to tibble
  tibble() %>%
  # update column names to follow snake case naming convention
  janitor::clean_names() %>%
  # correctly read excel dates as dates instead of numeric values
```

```
  mutate(across(.cols = c("dob"),
                .fns = janitor::excel_numeric_to_date))

# check to make sure that there are no duplicate people and that
# each row represents one person
person_data %>%
  count(person_id) %>%
  filter(n > 1)
```

```
## # A tibble: 0 x 2
## # i 2 variables: person_id <dbl>, n <int>
```

Exercise

Consider the year from July 1, 2021 to June 30, 2022 as the analysis period.

---

1. How many total bookings into the jail were there in that time period?

```
bookings_in_analysis_period <- incarceration_data %>%
  filter(date_in >= "2021-07-01" &
           date_in <= "2022-06-30")

nrow(bookings_in_analysis_period) %>%
  scales::comma()
```

```
## [1] "21,842"
```

---

2. How many unique people were booked into the jail?

```
bookings_in_analysis_period %>%
  select(person_id) %>%
  distinct() %>%
  count() %>%
  pull() %>%
  scales::comma()
```

```
## [1] "15,510"
```

---

3. How many people were in the jail at the moment of the data extraction?

This would include not only the people who were booked during the analysis period, but the people who were booked before analysis period, but were not yet released

```
# gather bookings where a person was booked before the analysis period, but
# has no release date
bookings_still_incarcerated <- incarceration_data %>%
  filter(date_in < "2021-07-01",
         is.na(release_out))

nrow(bookings_still_incarcerated) %>%
  scales::comma()
```

```
## [1] "130"
```

```
bookings_still_incarcerated %>%
  summarise(first_booking = min(date_in),
            last_booking = max(date_in))
```

```
## # A tibble: 1 x 2
##   first_booking last_booking
##   <date>        <date>
## 1 2016-09-07    2021-06-30
```

```
# gather bookings where a person was booked before the analysis period,
# but they were released within the analysis period
bookings_released_during_extraction <- incarceration_data %>%
  filter(date_in < "2021-07-01",
         release_out >= "2021-07-01" &
           release_out <= "2022-06-30")

nrow(bookings_released_during_extraction) %>%
  scales::comma()
```

```
## [1] "1,276"
```

```
bookings_released_during_extraction %>%
  summarise(first_release = min(release_out),
            last_release = max(release_out))
```

```
## # A tibble: 1 x 2
##   first_release last_release
##   <date>        <date>
## 1 2021-07-01    2022-06-29
```

```
# now create a table with all the bookings that we have identified to
# have occurred during the extraction and bookings that
# occurred before the extraction, but were not yet released
bookings_during_extraction <- bookings_in_analysis_period %>%
  bind_rows(bookings_still_incarcerated) %>%
```

```
  bind_rows(bookings_released_during_extraction)

# verify no duplicate bookings
bookings_during_extraction %>%
  count(booking_number) %>%
  filter(n > 1)
```

```
## # A tibble: 0 x 2
## # i 2 variables: booking_number <chr>, n <int>
```

```
# now count the number of people in jail at the moment of the
# data extraction
bookings_during_extraction %>%
  select(person_id) %>%
  distinct() %>%
  count() %>%
  pull() %>%
  scales::comma()
```

```
## [1] "16,456"
```

---

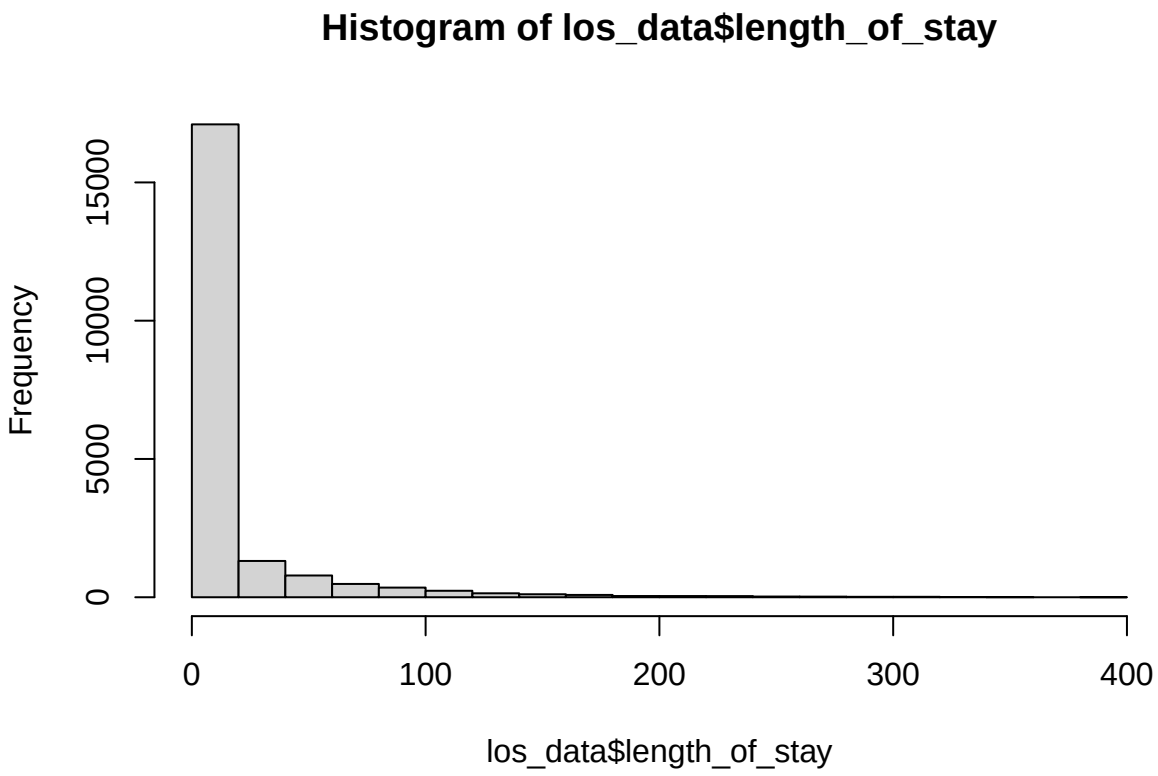
4. Consider the length of stay (LOS): the duration of each booking. Describe the LOS over the year analysis period. What insights (statistical and otherwise) does it provide you about variations in the jail population?

To look at length of stay during the analysis period, I think it is best to look at the bookings that were from July 1, 2021 to June 30, 2022. Overall, the data seems to suggest that the people who are detained in jail tend to be changing frequently, if half the jail population is only detained for 3 days or less.

- About 51.3% of bookings had a length of stay of at least 3 days.
- The most frequent length of stay being 2 day.
- There are about 25% of bookings that ranged from 10 days to 386.
- About 75.6% of people were only booked once

```
los_data <- bookings_in_analysis_period %>%
  # remove bookings with no release date
  filter(!is.na(release_out)) %>%
  # calculate length of stay
  mutate(length_of_stay = as.numeric(release_out - date_in) + 1)


# examine the distribution of the length of stay for this analysis period
hist(los_data$length_of_stay)
```




```
# get descriptive statistics to better understand the distribution
los_data %>%
  select(length_of_stay) %>%
  summary()
```

```
##  length_of_stay
##  Min.   :  1.00
##  1st Qu.:  2.00
##  Median :  3.00
##  Mean   : 16.54
##  3rd Qu.: 10.00
##  Max.   :386.00
```

```
los_data %>%
  tabyl(length_of_stay) %>%
  adorn_pct_formatting() %>%
  slice(1:21)
```

```
##  length_of_stay    n percent
##               1 2404   11.5%
##               2 6041   29.0%
##               3 2238   10.8%
##               4 1227    5.9%
##               5  867    4.2%
##               6  772    3.7%
```

```
##               7  734    3.5%
##               8  651    3.1%
##               9  477    2.3%
##              10  274    1.3%
##              11  220    1.1%
##              12  174    0.8%
##              13  181    0.9%
##              14  188    0.9%
##              15  155    0.7%
##              16  124    0.6%
##              17  114    0.5%
##              18   80    0.4%
##              19  114    0.5%
##              20   65    0.3%
##              21   90    0.4%
```

```r
los_data %>%
  mutate(grouped_los = ifelse(length_of_stay >= 10, "10+", length_of_stay),
         grouped_los = factor(grouped_los, levels = c(0:9, "10+"))) %>%
  tabyl(grouped_los) %>%
  adorn_pct_formatting()
```

```
##  grouped_los    n percent
##            0    0    0.0%
##            1 2404   11.5%
##            2 6041   29.0%
##            3 2238   10.8%
##            4 1227    5.9%
##            5  867    4.2%
##            6  772    3.7%
##            7  734    3.5%
##            8  651    3.1%
##            9  477    2.3%
##          10+ 5407   26.0%
```

```r
# how many times are people rebooked (booked at least one time)
rebooking <- los_data %>%
  arrange(person_id, date_in) %>%
  group_by(person_id) %>%
  mutate(frequency_booked = 1:n()) %>%
  ungroup() %>%
  arrange(person_id, desc(frequency_booked)) %>%
  distinct(person_id, .keep_all = T) %>%
  mutate(booked_multiple_times = ifelse(frequency_booked >= 2, "2+", "1"))

# review the total times people have been booked
rebooking %>%
  tabyl(frequency_booked) %>%
  adorn_pct_formatting()
```

```
##  frequency_booked     n percent
##                 1 11348   75.6%
##                 2  2391   15.9%
```

```
##                  3   774    5.2%
##                  4   298    2.0%
##                  5   109    0.7%
##                  6    54    0.4%
##                  7    23    0.2%
##                  8     7    0.0%
##                  9     5    0.0%
##                 13     1    0.0%
##                 14     1    0.0%
##                 16     1    0.0%
```

```
# review how many people have been booked more than once
rebooking %>%
  tabyl(booked_multiple_times) %>%
  adorn_pct_formatting()
```

```
##  booked_multiple_times     n percent
##                      1 11348   75.6%
##                     2+  3664   24.4%
```

Additionally, the length of stay is positively skewed so if we wanted to examine what the median length of stay was for this population with 95% certainty, a survival analysis would make the most sense. In this scenario, we would be trying to understand the amount of time it takes a person who was booked to be released from jail during our analysis period.

```
# create the event flag and create a time flag that represents length of stay
# remembering that this length of stay is until the end of the study period
survival_data <- bookings_in_analysis_period %>%
  mutate(event = case_when(
    is.na(release_out) ~ 0,
    release_out > "2022-06-30" ~ 0,
    T ~ 1)) %>%
  mutate(time = case_when(
    is.na(release_out) ~ as.numeric(ymd("2022-06-30") - date_in) + 1,
    release_out > "2022-06-30" ~ as.numeric(ymd("2022-06-30") - date_in) + 1,
    T ~ as.numeric(release_out - date_in) + 1
  )) %>%
  select(booking_number, time, event)

# check the number of events (releases) that occurred during the analysis period
survival_data %>%
  tabyl(event) %>%
  adorn_pct_formatting()
```

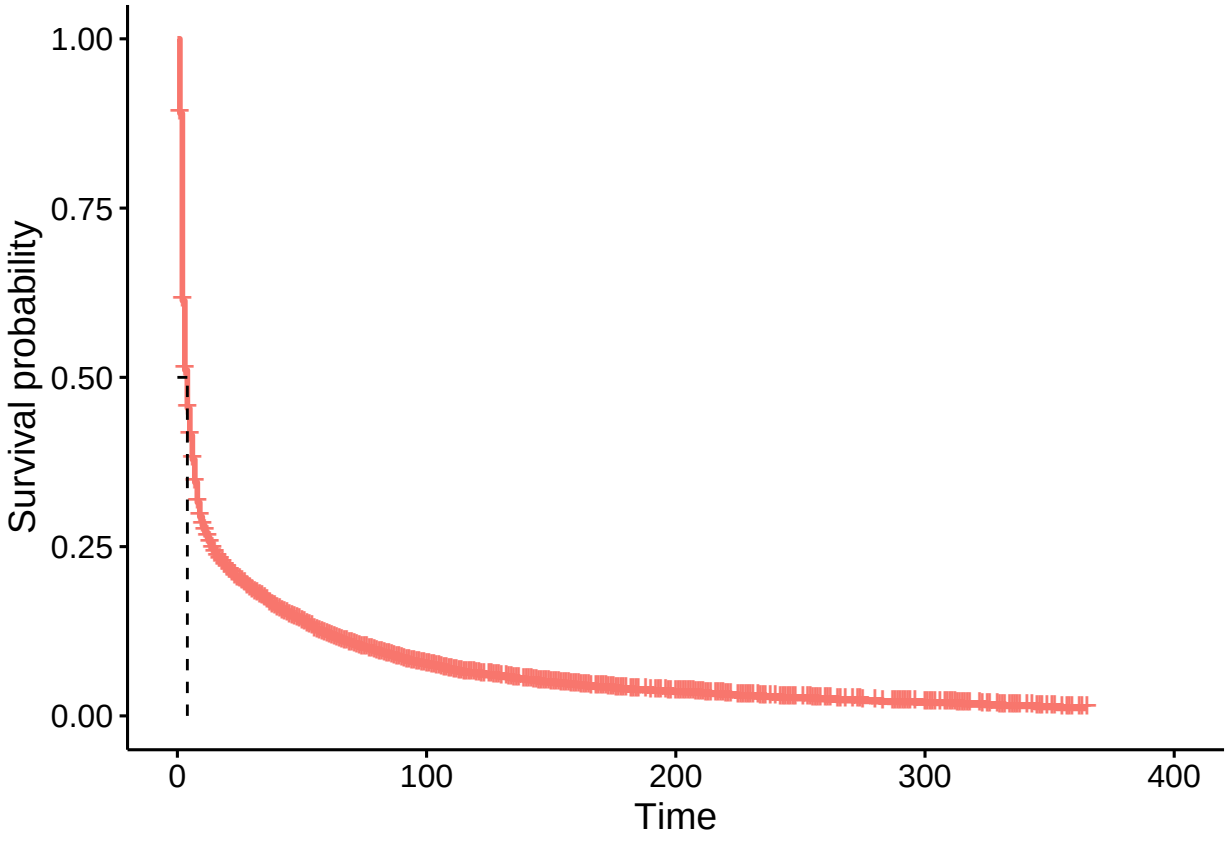
```
##  event     n percent
##      0  1536    7.0%
##      1 20306   93.0%
```

```
# calculate the Kaplan-Meier estimate
km <- survival::survfit(Surv(time, event) ~ 1,
  data = survival_data
)

km
```

```
## Call: survfit(formula = Surv(time, event) ~ 1, data = survival_data)
##
##          n events median 0.95LCL 0.95UCL
## [1,] 21842  20306      4       4       4
```

```
# examine a graph of the probablity of release
survminer::ggsurvplot(km,
  conf.int = FALSE,
  surv.median.line = "hv",
  legend = "none"
)
```



From this analysis, the median release time for a person can be estimated to be 4 days, with a 95% CI [4,4]. One thing to note is that this median does differ from our review of the discriptive statistics because in this dataset we kept bookings with a missing date and whereas these bookings were excluded previous. A next step would be to examine if an individual being booked again during the study period would impact their length of stay.

5. What was the average daily population in the jail during that year? Daily population ought to include anyone who spent even one minute in the jail in a given day. Please describe the methods/approach you used to answer this question. What tool did you use? What functions or other capabilities within the tool? (That is, help another analyst replicate what you did. Sharing code with your answers is encouraged but by no means required.)

Use a for loop to create a dataset where if a booking occurred from monday to friday (5 days) the booking will be spread across 5 rows. This will create a dataset where we can look at who was in the jail each day of the analysis period. In order to correctly capture all the individuals in the jail during this time, we will use the dataset `bookings_during_extraction` created in question 3.

```
# initialize table
booking_long = NULL

for(ith_booking in 1:nrow(bookings_during_extraction)){

  # gather the current bookings information
  booking = bookings_during_extraction[ith_booking, ]

  # if release date is missing, set this to the last day of the analysis period
  # to capture each individual, even people who are still in jail after the analysis time constraint
  if(!is.na(booking$release_out)) {
    release_out <- booking$release_out
  } else {
     release_out = as.Date("2022-06-30")
  }

  # convert the booking row into a long table where each row represents a day
  # the person was in jail based on their current booking.
  dates_incarcerated = seq(booking$date_in, release_out, by = "1 day")
  ith_booking_long <- tibble(date = dates_incarcerated,
                             booking_number = booking$booking_number,
                             person_id = booking$person_id)

  # combine the current bookings table with the larger dataset.
  booking_long <- booking_long %>%
    bind_rows(ith_booking_long)
}
```

Now that we have a dataset where a bookings length of stay is spreadout by date, we can look at the daily population.

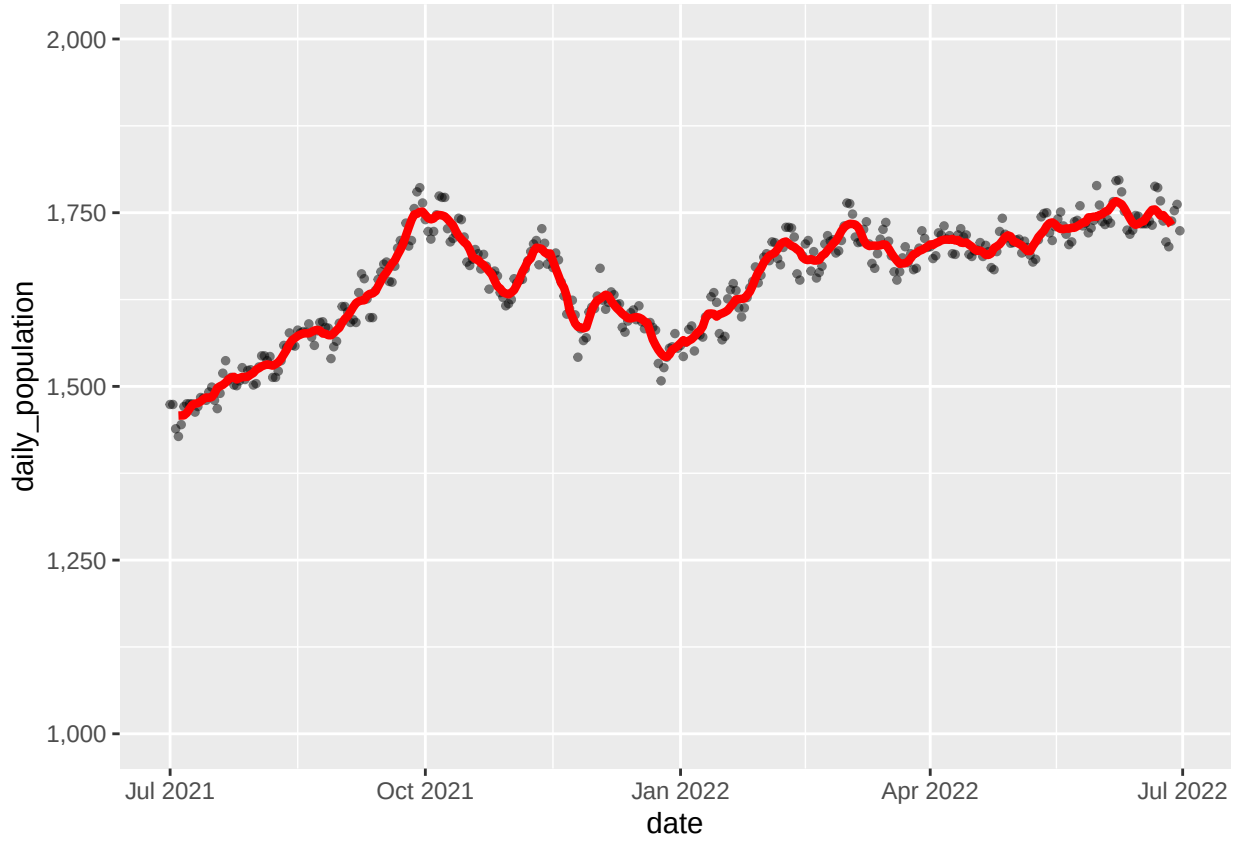
```
# count the number of people in jail each day
bookings_by_day <- booking_long %>%
  filter(date >= "2021-07-01" & date <= "2022-06-30") %>%
  arrange(date) %>%
  group_by(date) %>%
  count() %>%
  ungroup() %>%
  rename(daily_population = n)

# What is the average daily population for the analysis period?
(average_daily_population <- mean(bookings_by_day$daily_population))
```

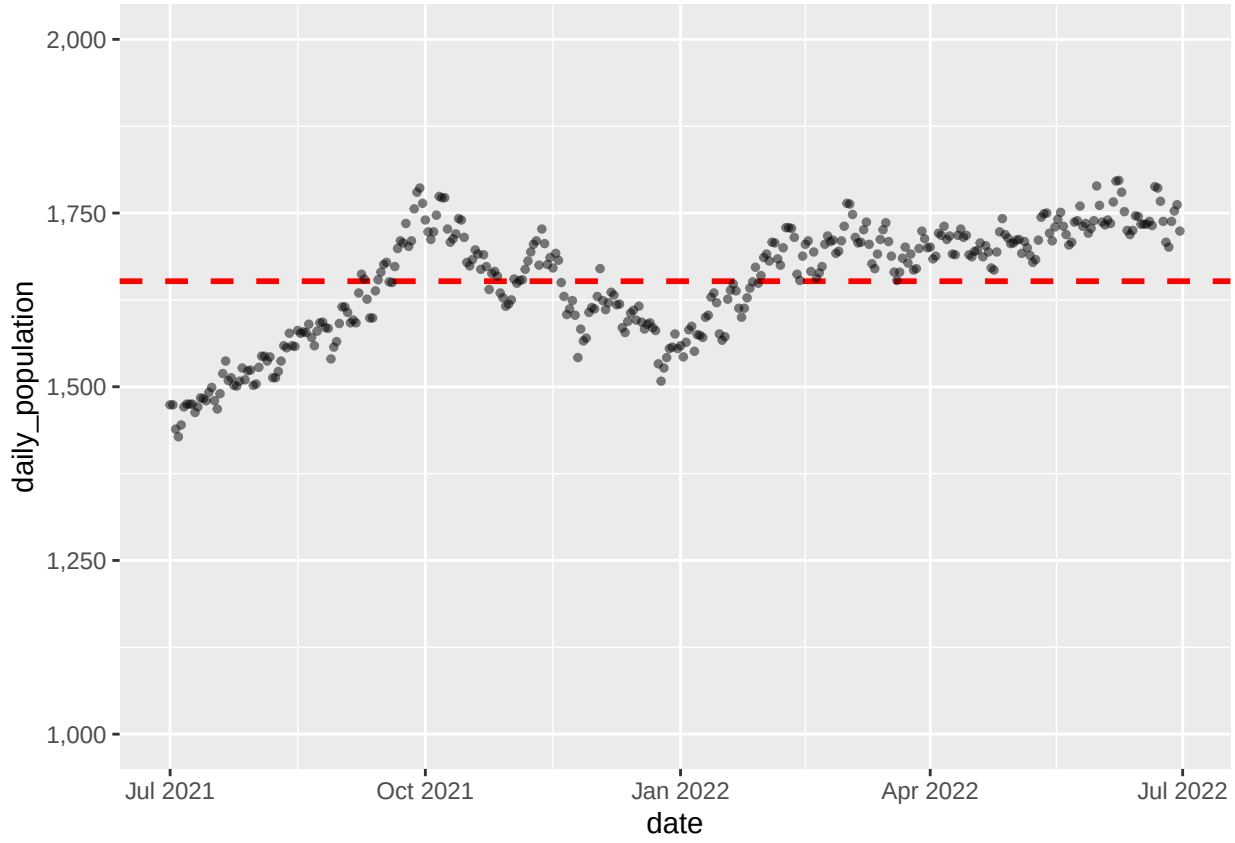
```
## [1] 1651.923
```

From the calculations above, we find that the average daily population in the jail during our analysis period is 1,652

```
# what does the daily population look like overtime? note using a moving average
# to smooth the estimates a little
bookings_by_day %>%
  mutate(seven_day_moving_average = zoo::rollmean(daily_population, k = 7, fill = NA)) %>%
  ggplot(aes(x = date, y = daily_population)) +
  geom_point(alpha = 0.5, size = 1) +
  geom_line(aes(x = date, y = seven_day_moving_average), color = "red", size = 1.5) +
  scale_y_continuous(limit = c(1000, 2000), labels = comma)
```




```
# View the daily population against the average daily population. Around
# what months is the daily population greater than the average?
bookings_by_day %>%
  ggplot(aes(x = date, y = daily_population)) +
  geom_hline(aes(yintercept = average_daily_population), color = "red", size = 1, linetype = "dashed") +
  geom_point(alpha = 0.5, size = 1) +
  scale_y_continuous(limit = c(1000, 2000), labels = comma)
```



---

6. Which day during that year had the lowest daily population? What was the population that day? Which day had the highest daily population? What was the population that day?

```
# date with the lowest daily population
bookings_by_day %>%
  filter(daily_population == min(daily_population))
```

```
## # A tibble: 1 x 2
##   date       daily_population
##   <date>                <int>
## 1 2021-07-04             1428
```

```
# date with the highest daily  population
bookings_by_day %>%
  filter(daily_population == max(daily_population))
```

```
## # A tibble: 1 x 2
##   date       daily_population
##   <date>                <int>
## 1 2022-06-08             1797
```

---

7. Please provide a basic analysis of jail demographics during that year period. Are there meaningful statistical relationships among the different groups in the jail?

For this analysis, we are assuming that each booking is a new person, where people who have been booked more than once will be double counted.

- The majority of the jail population was men (77%) where their length of stay seemed longer and varied more compared to women.
- A large proportion of the jail population was white (71%) and the variance of length of stay among the other race's seemed consistent, with the exception of the Asian or Pacific Islander group.
- The age of the people in jail during this time period was also positively skewed where the jail population tends to be younger. At least 50% of the population was 33 years old and younger.

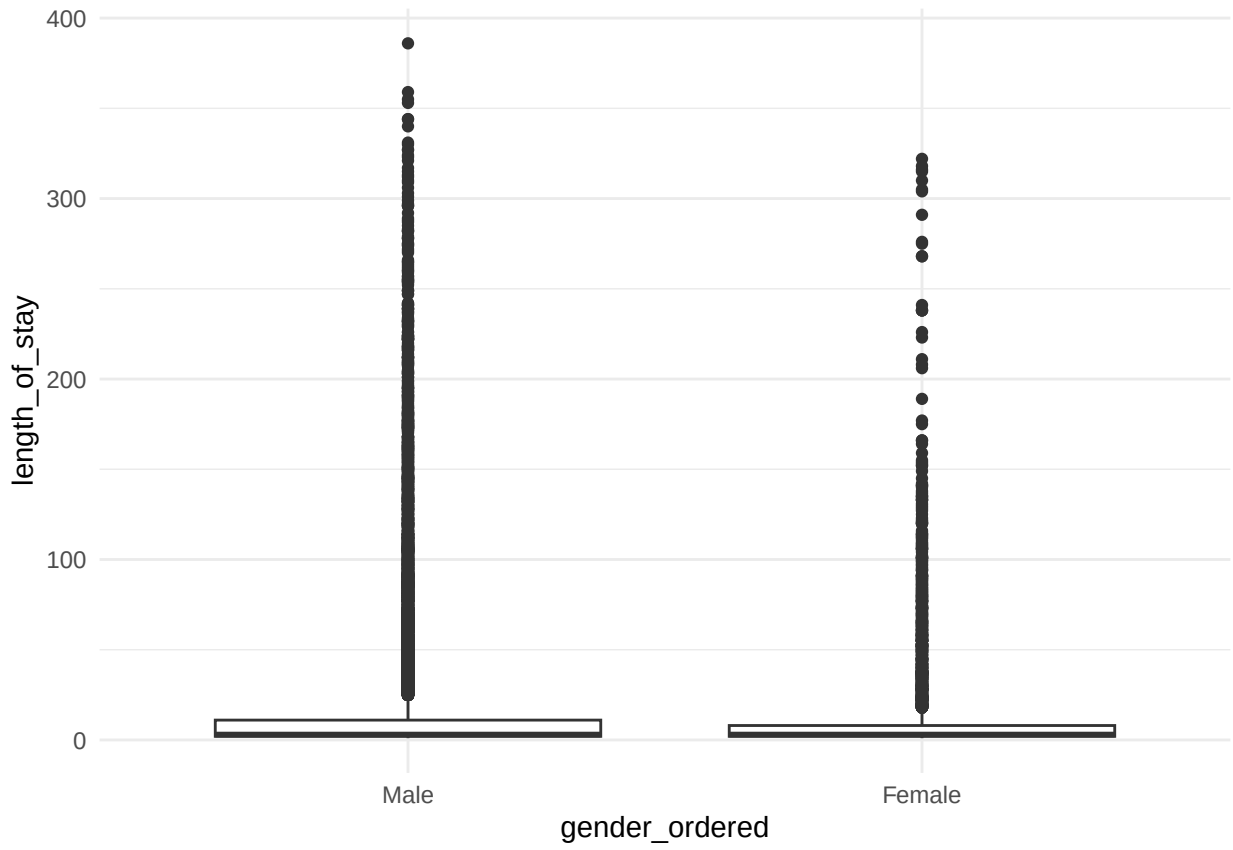
```
booking_demographics <- bookings_in_analysis_period %>%
  # remove people with a missing release date
  filter(!is.na(release_out)) %>%
  # add in demographic data
  left_join(person_data) %>%
  mutate(age_at_booking = floor(interval(dob, date_in) / years(1))) %>%
  # calculate length of stay
  mutate(length_of_stay = as.numeric(release_out - date_in) + 1) %>%
  mutate(gender_ordered = factor(gender, levels = c("Male", "Female")))
```

```
## Joining with `by = join_by(person_id)`
```

```
# descriptive stats by gender & length of stay distribution
booking_demographics %>%
  group_by(gender_ordered) %>%
  summarise(n = n(),
            median_los = median(length_of_stay),
            los_iqr = IQR(length_of_stay)) %>%
  ungroup() %>%
  mutate(total = sum(n),
         pct_of_total = scales::percent(n/total)) %>%
  select(gender_ordered, n, pct_of_total, median_los, los_iqr) %>%
  arrange(n)
```

```
## # A tibble: 2 x 5
##   gender_ordered     n pct_of_total median_los los_iqr
##   <fct>          <int> <chr>              <dbl>   <dbl>
## 1 Female          4690 23%                    3       6
## 2 Male           16128 77%                    3       9
```

```
booking_demographics %>%
  ggplot(aes(x = length_of_stay, y = gender_ordered)) +
  geom_boxplot() +
  coord_flip() +
  theme_minimal()
```

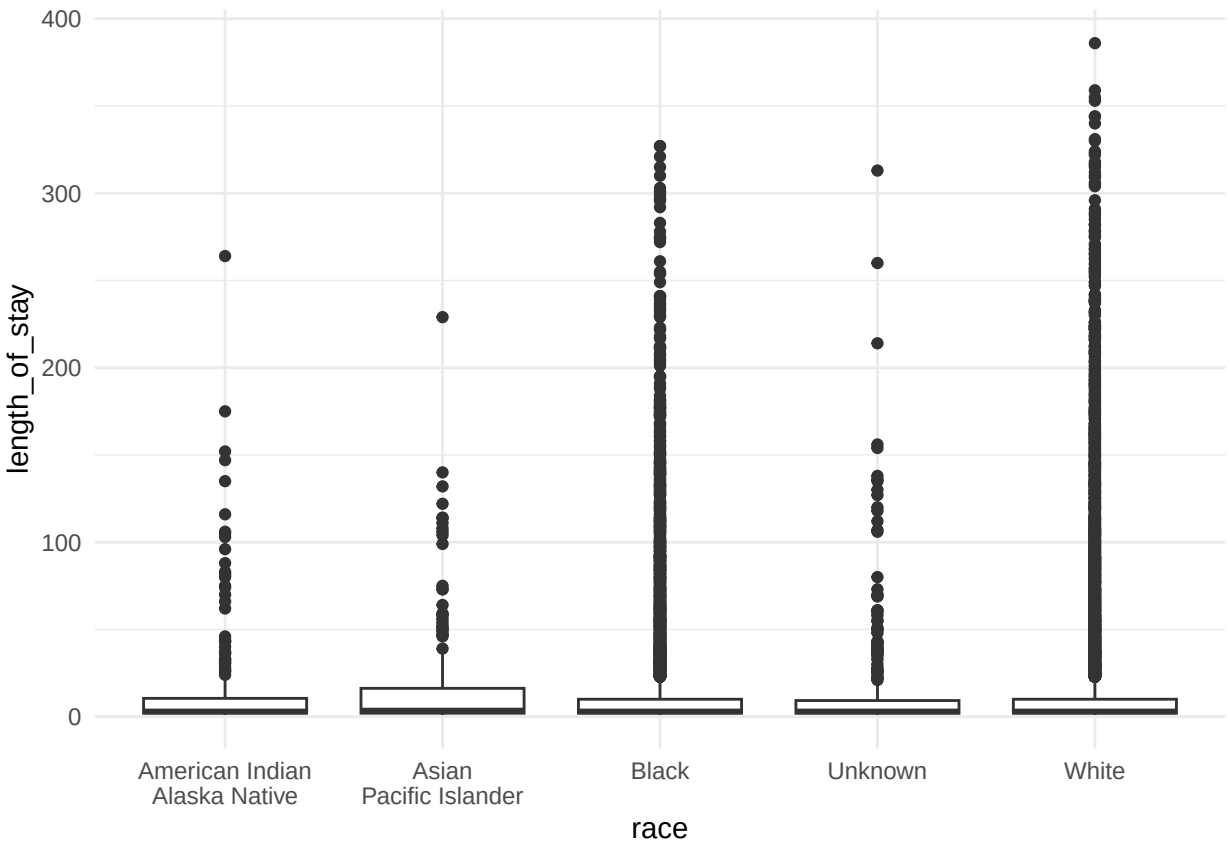



```
# descriptive stats by race & length of stay distribution
booking_demographics %>%
  group_by(race) %>%
  summarise(n = n(),
            median_los = median(length_of_stay),
            los_iqr = IQR(length_of_stay)) %>%
  ungroup() %>%
  mutate(total = sum(n),
         pct_of_total = scales::percent(n/total)) %>%
  select(race, n, pct_of_total, median_los, los_iqr) %>%
  arrange(n)
```

```
## # A tibble: 5 x 5
##   race                                n pct_of_total median_los los_iqr
##   <chr>                           <int> <chr>              <dbl>   <dbl>
## 1 Asian or Pacific Islander         172 0.83%                3.5   14.2
## 2 American Indian or Alaska Native  259 1.24%                3      8.5
## 3 Unknown                           372 1.79%                3      7.25
## 4 Black                            5241 25.18%               3      8
## 5 White                           14774 70.97%               3      8
```

```
booking_demographics %>%
  mutate(race = addline_format(race)) %>%
  ggplot(aes(x = length_of_stay, y = race)) +
  geom_boxplot() +
```

```
  coord_flip() +
  theme_minimal()
```

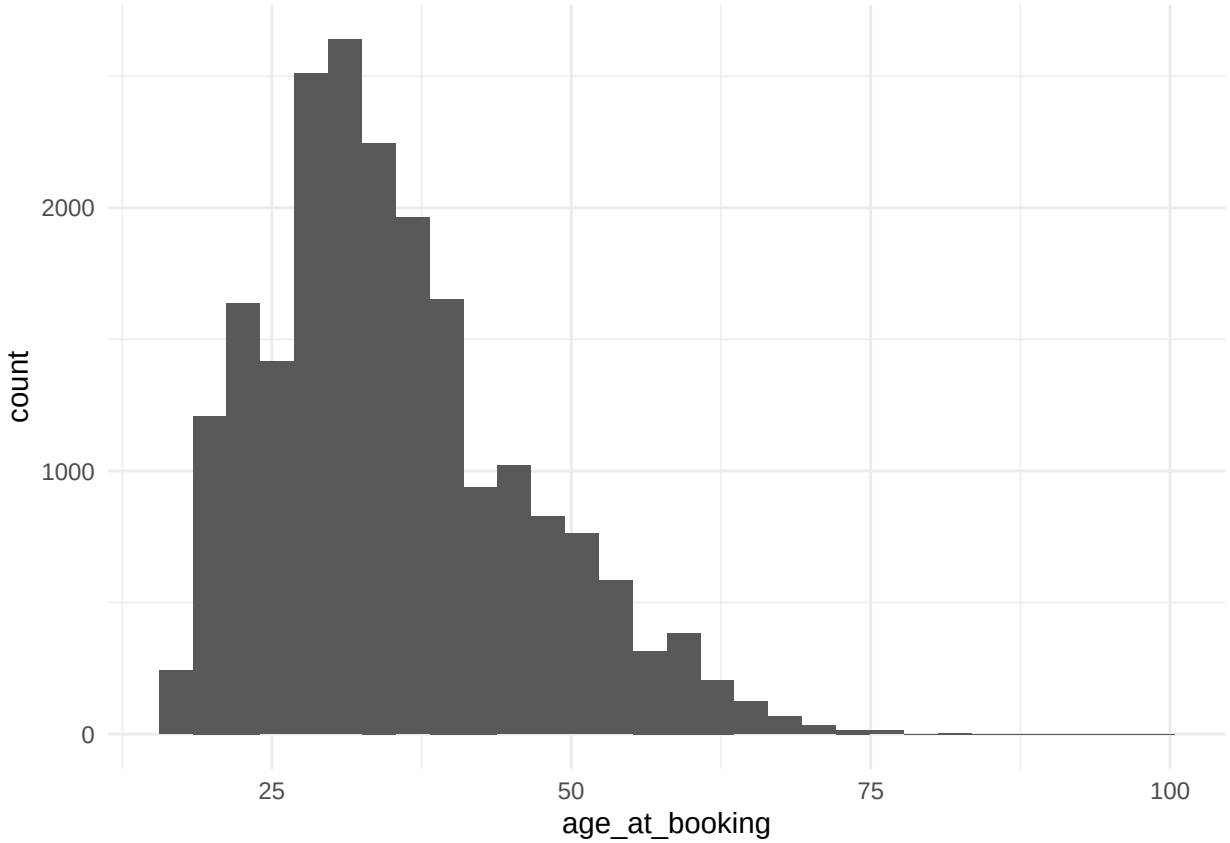



```
# descriptive stats by age
booking_demographics %>%
  filter(!is.na(age_at_booking)) %>%
  summarise(median_age = median(age_at_booking),
            mean_age = mean(age_at_booking)) %>%
  ungroup() %>%
  select(median_age, mean_age)
```

```
## # A tibble: 1 x 2
##   median_age mean_age
##        <dbl>    <dbl>
## 1         33     35.4
```

```
booking_demographics %>%
  ggplot(aes(x = age_at_booking)) +
  geom_histogram() +
  theme_minimal()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Because of how skewed the data is for length of stay amoung each demographic, I believe that Wilcoxon rank sum test would be the best way to examine the median by demographic and see if there is a difference between groups. To examine the median for race, people who are white are going to be the control group.

```
# is the median length of stay greater for male detainees than women detainees
wilcox.test(formula = length_of_stay ~ gender_ordered,
            data = booking_demographics,
            extact = FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  length_of_stay by gender_ordered
## W = 39629454, p-value = 4.084e-07
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(formula = length_of_stay ~ gender_ordered,
            data = booking_demographics,
            extact = FALSE,
            alternative = "greater")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  length_of_stay by gender_ordered
## W = 39629454, p-value = 2.042e-07
## alternative hypothesis: true location shift is greater than 0
```

```r
# is the median length of stay less for white than other races?

## Asian or Pacific Islander
white_and_asian_pacific_islander <- booking_demographics %>%
  filter(race == "White" |
           race == "Asian or Pacific Islander") %>%
  mutate(race_ordered = factor(race, levels = c("White", "Asian or Pacific Islander")))

wilcox.test(formula = length_of_stay ~ race_ordered,
            data = white_and_asian_pacific_islander,
            extact = FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  length_of_stay by race_ordered
## W = 1210675, p-value = 0.2806
## alternative hypothesis: true location shift is not equal to 0
```

```r
## American Indian or Alaska Native
white_and_american_indian <- booking_demographics %>%
  filter(race == "White" |
           race == "American Indian or Alaska Native") %>%
  mutate(race_ordered = factor(race, levels = c("White", "American Indian or Alaska Native")))

wilcox.test(formula = length_of_stay ~ race_ordered,
            data = white_and_american_indian,
            extact = FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  length_of_stay by race_ordered
## W = 2063379, p-value = 0.0279
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(formula = length_of_stay ~ race_ordered,
            data = white_and_american_indian,
            extact = FALSE, alternative = "greater")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  length_of_stay by race_ordered
## W = 2063379, p-value = 0.01395
## alternative hypothesis: true location shift is greater than 0
```

```r
## Unknown
white_and_unknown <- booking_demographics %>%
  filter(race == "White" |
           race == "Unknown") %>%
  mutate(race_ordered = factor(race, levels = c("White", "Unknown")))
```

```r
wilcox.test(formula = length_of_stay ~ race_ordered,
            data = white_and_unknown,
            extact = FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  length_of_stay by race_ordered
## W = 2798769, p-value = 0.5363
## alternative hypothesis: true location shift is not equal to 0
```

```r
## Black
white_and_black <- booking_demographics %>%
  filter(race == "White" |
           race == "Black") %>%
  mutate(race_ordered = factor(race, levels = c("White", "Black")))

wilcox.test(formula = length_of_stay ~ race_ordered,
            data = white_and_black,
            extact = FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  length_of_stay by race_ordered
## W = 38437783, p-value = 0.4337
## alternative hypothesis: true location shift is not equal to 0
```

The Wilcoxon test showed that the medians were significantly different ($p<0.0001$) between men and women in our jail population where the median length of stay for women was 3 (IRQ = 6) and the median length of stay among men was 3 (IQR = 9). An additional Wilcoxon test showed that the difference median was significantly different ($p<0.0279$) between the white jail detainees who had a median length of stay of 3 (IQR = 8) and American Indian or Alaska Natives who had a median length of stay of 3 (IR = 8.5).

---

8. What other insights, useful facts, or questions/concerns did you uncover, if any, in the data?

- While working with this data, I was curious about what others have noticed when it comes to length of stay amoung people who are detained. One similarlity that I noticed between this data and the peer reviewed papers that I found published was how young people were when detained. In the paper that I found, a relationship the researchers noticed amoung their cohort was that the length of stay was longer for people who were younger. I would love to investigate this further and see if this relationship is seen in this study population.
- An additional research question I found while getting familiar with this topic was the relationship to length of stay and being detained more than once. Among the bookings in this analysis period, 24.4% of people were detained more than once so I wonder how this impacted an individuals second stay. This could be examined using a survival analysis.

- Due to the large population of bookings with length of stays that were 10 days or longer (26%), I would like to examine the demographic differences betwen people with a short, medium, and long length of stay. An analysis that looked at these 3 categories might give us more insight to the relationship between demographics and length of stay.
- Lastly, I would like to make a visual that compares the number of people detained by race in our analysis year to the general ppopulation and maybe narrow this down to the state or county the jail is located to see if the population in the jail mirrors or is different to the general public. An example of the graphic I am thinking about can be found here: https://stephanieevergreen.com/proportion-plots/