STA2453 Lab 1

Yihan Duan

21/09/2021

Exercise 1

Data Quality

Load raw ED data.

```r
knitr::opts_chunk$set(echo = TRUE)

# Load the required libraries
library(tidyverse)
library(lubridate)

# Read in the data
ed_data <- read_csv("raw_ed_data.csv")

# Show head
head(ed_data)
```

```
## # A tibble: 6 x 10
##   ENCOUNTER_NUM CTAS_CD CTAS_DESCR    ed_start_time       ed_end_time
##           <dbl> <chr>   <chr>         <dttm>              <dttm>
## 1             1 2       EMERGENCY     2019-01-01 06:06:00 2019-01-01 09:56:00
## 2             2 3       URGENT        2019-01-01 06:11:00 2019-01-01 06:43:00
## 3             3 3       URGENT        2019-01-01 06:21:00 2019-01-01 11:27:00
## 4             4 2       EMERGENCY     2019-01-01 06:36:00 2019-01-01 11:48:00
## 5             5 1       RESUSCITATION 2019-01-01 06:37:00 2019-01-01 08:27:00
## 6             6 2       EMERGENCY     2019-01-01 06:42:00 2019-01-01 11:48:00
## # ... with 5 more variables: ed_pia_time <dttm>, adm_start_time <dttm>,
## #   admitted <dbl>, los <dbl>, presenting_complaint <chr>
```

Clean up 'presenting_complaint' column as shown in class.

```r
# function to clean presenting complaints text
clean_complaints <- function(x) {

    x_clean <- x %>%
        # remove any leading and trailing spaces
    trimws() %>%
        # collapse > 1 blank space into 1 blank space
    gsub(" +", " ", .) %>%
        # set text to lower case
    tolower()

    return(x_clean)
}

ed_data <- ed_data %>%
    mutate(presenting_complaint = clean_complaints(presenting_complaint))

ed_data <- ed_data %>%
    mutate(presenting_complaint = case_when(presenting_complaint == "chest pian" ~
        "chest pain", presenting_complaint == "burns" ~ "burn", presenting_complaint ==
        "traumatic injuries" ~ "traumatic injury", presenting_complaint %in% c("unk",
        "missing") ~ "unknown", presenting_complaint == "headach" ~ "headache", TRUE ~
        presenting_complaint))

ed_data %>%
    count(presenting_complaint, sort = T) %>%
    mutate(proportion = round(n/sum(n), 3))
```

```
## # A tibble: 17 x 3
##    presenting_complaint       n proportion
##    <chr>                  <int>      <dbl>
##  1 abdominal pain         14508      0.18
##  2 sore throat            13735      0.171
##  3 loss of hearing         8069      0.1
##  4 confusion               7742      0.096
##  5 headache                6979      0.087
##  6 upper extremity injury  4137      0.051
##  7 lower extremity injury  4077      0.051
##  8 back pain               3607      0.045
##  9 rash                    3490      0.043
## 10 chest pain              2767      0.034
## 11 general weakness        2764      0.034
## 12 traumatic injury        2610      0.032
## 13 hallucinations          2080      0.026
## 14 bizarre behaviour       1595      0.02
## 15 burn                    1232      0.015
## 16 trouble breathing        821      0.01
## 17 unknown                  251      0.003
```

As shown in class, we can count all the 'NA's for each column.

```
count_NAs <- function(x) {

    num_NAs <- sum(is.na(x))

    return(num_NAs)
}

# the data
ed_data %>%
    # becomes the first argument passed to the summarize_all function
summarize_all(count_NAs) %>%
    glimpse
```

```
## Rows: 1
## Columns: 10
## $ ENCOUNTER_NUM        <int> 0
## $ CTAS_CD              <int> 0
## $ CTAS_DESCR           <int> 0
## $ ed_start_time        <int> 793
## $ ed_end_time          <int> 396
## $ ed_pia_time          <int> 0
## $ adm_start_time       <int> 68849
## $ admitted             <int> 0
## $ los                  <int> 1186
## $ presenting_complaint <int> 0
```

```
ed_data %>%
    filter(is.na(ed_start_time) & is.na(ed_end_time))
```

```
## # A tibble: 3 x 10
##   ENCOUNTER_NUM CTAS_CD CTAS_DESCR    ed_start_time ed_end_time
##           <dbl> <chr>   <chr>         <dttm>        <dttm>
## 1         15461 4       SEMI-URGENT   NA            NA
## 2         29442 1       RESUSCITATION NA            NA
## 3         71909 3       URGENT        NA            NA
## # ... with 5 more variables: ed_pia_time <dttm>, adm_start_time <dttm>,
## #   admitted <dbl>, los <dbl>, presenting_complaint <chr>
```

Now we check all the variables one by one.

ENCOUNTER_NUM

First see if there are any duplicate numbers.

```
print("Length of ENCOUNTER_NUM:")
```

```
## [1] "Length of ENCOUNTER_NUM:"
```

```
length(ed_data$ENCOUNTER_NUM)
```

```
## [1] 80464
```

```
print("Number of unique values of ENCOUNTER_NUM:")
```

```
## [1] "Number of unique values of ENCOUNTER_NUM:"
```

```
length(unique(ed_data$ENCOUNTER_NUM))
```

```
## [1] 80248
```

We can see that there are duplicates in ENCOUNTER_NUM column.

```
# Remove all duplicate rows
ed_data_dedup <- ed_data[!duplicated(ed_data), ]
```

Now check the number of unique values again.

```
print("Length of ENCOUNTER_NUM:")
```

```
## [1] "Length of ENCOUNTER_NUM:"
```

```
length(ed_data_dedup$ENCOUNTER_NUM)
```

```
## [1] 80249
```

```
print("Number of unique values of ENCOUNTER_NUM:")
```

```
## [1] "Number of unique values of ENCOUNTER_NUM:"
```

```
length(unique(ed_data_dedup$ENCOUNTER_NUM))
```

```
## [1] 80248
```

Still one duplicate ENCOUNTER_NUM, lets find it.

```
num_freq <- ed_data_dedup %>%
    count(ENCOUNTER_NUM) %>%
    filter(n > 1)
dup_num = num_freq$ENCOUNTER_NUM[1]
ed_data_dedup %>%
    filter(ENCOUNTER_NUM == dup_num)
```

```
## # A tibble: 2 x 10
##   ENCOUNTER_NUM CTAS_CD CTAS_DESCR ed_start_time       ed_end_time
##           <dbl> <chr>   <chr>      <dttm>              <dttm>
## 1         44042 2       EMERGENCY  2019-07-18 16:38:00 2019-07-19 06:02:00
## 2         44042 2       EMERGENCY  2019-07-18 16:38:00 2019-07-19 06:02:00
## # ... with 5 more variables: ed_pia_time <dttm>, adm_start_time <dttm>,
## #   admitted <dbl>, los <dbl>, presenting_complaint <chr>
```

Notice the only difference between these two records are the `ed_pia_time`.

CTAS_CD

Check data integrity for column 'CTAS_CD'.

```
ed_data_dedup %>%
    count(CTAS_CD)
```

```
## # A tibble: 6 x 2
##   CTAS_CD     n
##   <chr>   <int>
## 1 1        3289
## 2 2       26029
## 3 3       34688
## 4 4       12009
## 5 5        3033
## 6 N/A      1201
```

As we can see, the values presented in this column are mostly in range [1, 5]. However, there are many records with CTAS_CD missing.

CTAS_DESCR

First check all the values presented in CTAS_DESCR.

```
ed_data_dedup %>%
    count(CTAS_DESCR)
```

```
## # A tibble: 6 x 2
##   CTAS_DESCR        n
##   <chr>         <int>
## 1 EMERGENCY     26029
## 2 N/A            1201
## 3 NON URGENT     3033
## 4 RESUSCITATION  3289
## 5 SEMI-URGENT   12009
## 6 URGENT        34688
```

There should be a 1 to 1 mapping from CTAS_CD to CTAS_DESCR.

```
ed_data[, c("CTAS_CD", "CTAS_DESCR")] %>%
    unique
```

```
## # A tibble: 6 x 2
##   CTAS_CD CTAS_DESCR
##   <chr>   <chr>
## 1 2       EMERGENCY
## 2 3       URGENT
## 3 1       RESUSCITATION
## 4 N/A     N/A
## 5 4       SEMI-URGENT
## 6 5       NON URGENT
```

Yes the mapping is 1 to 1.

ed_start_time, ed_end_time, ed_pia_time, adm_start_time

ed_start_time should always come before ed_end_time.

```
print("PIA before arrival at the ED:")
```

```
## [1] "PIA before arrival at the ED:"
```

```
ed_data %>%
    filter(ed_start_time > ed_pia_time) %>%
    nrow
```

```
## [1] 16
```

```
print("Departure before arrival at the ED:")
```

```
## [1] "Departure before arrival at the ED:"
```

```
ed_data %>%
    filter(ed_start_time > ed_end_time) %>%
    nrow
```

```
## [1] 1591
```

```
print("PIA after leaving the ED:")
```

```
## [1] "PIA after leaving the ED:"
```

```
ed_data %>%
    filter(ed_pia_time > ed_end_time) %>%
    nrow
```

```
## [1] 3934
```

```
print("PIA after admitted to the hospital:")
```

```
## [1] "PIA after admitted to the hospital:"
```

```
ed_data %>%
    filter(ed_pia_time > adm_start_time) %>%
    nrow
```

```
## [1] 441
```

```r
print("Admitted to the hospital before leaving the ED:")
```

```
## [1] "Admitted to the hospital before leaving the ED:"
```

```r
ed_data %>%
    filter(ed_end_time > adm_start_time) %>%
    nrow
```

```
## [1] 11297
```

```r
print("Arrived at the ED after admitted to the hospital:")
```

```
## [1] "Arrived at the ED after admitted to the hospital:"
```

```r
ed_data %>%
    filter(ed_start_time > adm_start_time) %>%
    nrow
```

```
## [1] 7
```

```r
print("PIA time of '2099-01-01':")
```

```
## [1] "PIA time of '2099-01-01':"
```

```r
ed_data_dedup %>%
    filter(ed_pia_time == ymd("2099-01-01")) %>%
    nrow
```

```
## [1] 1582
```

```r
ed_data_dedup %>%
    filter(ed_pia_time == ymd("2099-01-01"))
```

```
## # A tibble: 1,582 x 10
##    ENCOUNTER_NUM CTAS_CD CTAS_DESCR ed_start_time       ed_end_time
##            <dbl> <chr>   <chr>      <dttm>              <dttm>
##  1           208 3       URGENT     2019-01-02 08:36:00 2019-01-02 21:32:00
##  2           228 3       URGENT     2019-01-02 13:26:00 2019-01-02 14:49:00
##  3           340 2       EMERGENCY  2019-01-02 21:00:00 2019-01-06 05:33:00
##  4           364 2       EMERGENCY  2019-01-02 22:33:00 2019-01-03 07:45:00
##  5           441 3       URGENT     2019-01-03 09:52:00 2019-01-03 19:24:00
##  6           478 2       EMERGENCY  2019-01-03 15:59:00 2019-01-03 17:52:00
##  7           529 3       URGENT     2019-01-03 19:29:00 2019-01-03 21:18:00
##  8           562 3       URGENT     2019-01-03 21:28:00 2019-01-04 01:48:00
##  9           598 2       EMERGENCY  2019-01-03 23:42:00 2019-01-04 10:00:00
## 10           641 2       EMERGENCY  2019-01-04 04:26:00 2019-01-03 14:35:00
## # ... with 1,572 more rows, and 5 more variables: ed_pia_time <dttm>,
## #   adm_start_time <dttm>, admitted <dbl>, los <dbl>,
## #   presenting_complaint <chr>
```

As we can see, many ‘ed_pia_time’ entries are labeled with unrealistic dates – ‘2099-01-01’.

adm_start_time, admitted

adm_start_time only make sense if the patient is admitted.

```r
ed_data %>%
    filter(admitted == 0) %>%
    count(adm_start_time)
```

```
## # A tibble: 1 x 2
##   adm_start_time     n
##   <dttm>         <int>
## 1 NA             68848
```

All patients that are not admitted have no ‘adm_start_tme’.

```
ed_data %>%
    filter(admitted == 1) %>%
    count_NAs()
```

```
## [1] 371
```

There are 371 records that are admitted but have no ‘adm_start_time’.

los

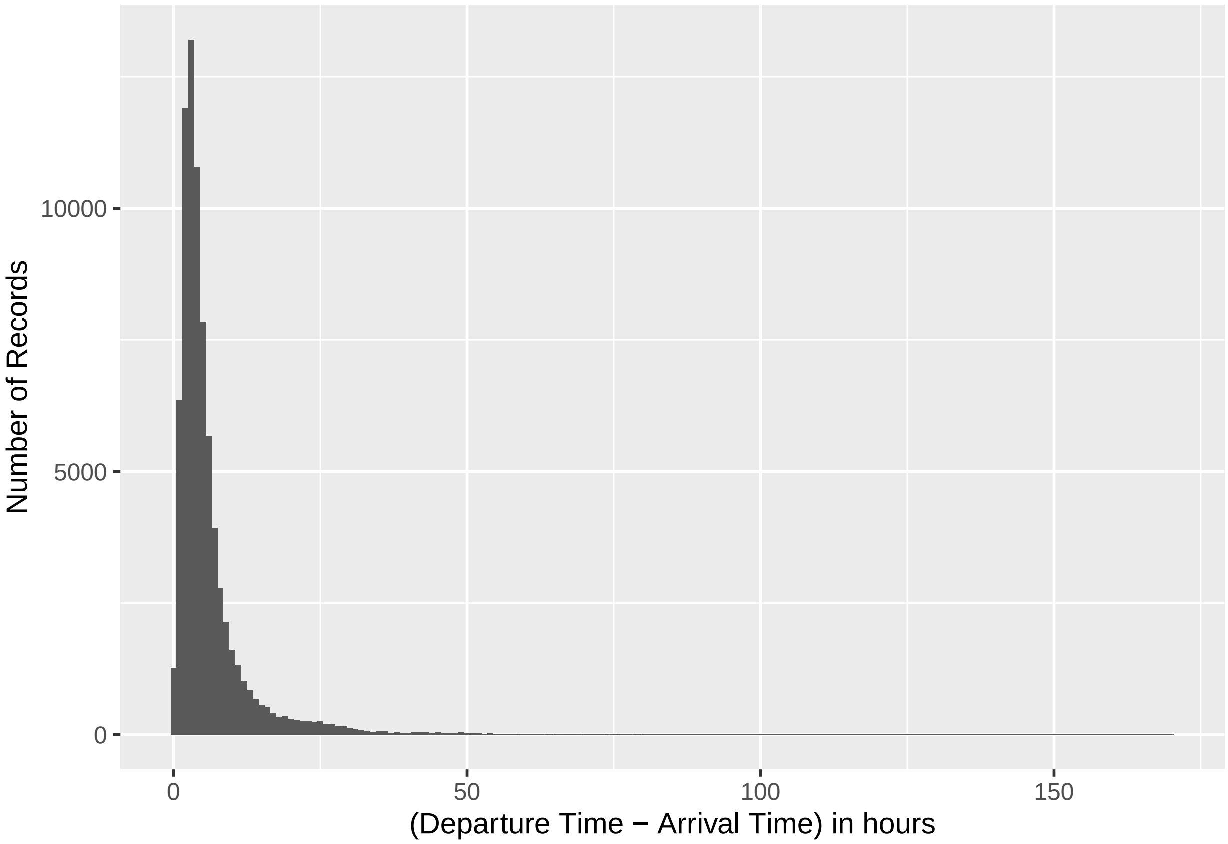
```
ed_data %>%
    filter(los == 24) %>%
    count
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  2772
```

Descriptive analysis

Length of stay

```
ed_data_dedup$startToEnd = as.numeric(difftime(ed_data_dedup$ed_end_time, ed_data_dedup$ed_start_time),
    units = "hours")

ed_data_dedup %>%
    filter(startToEnd >= 0) %>%
    ggplot(aes(x = startToEnd)) + geom_histogram(binwidth = 1) + labs(x = "(Departure Time - Arrival Ti
    y = "Number of Records")
```



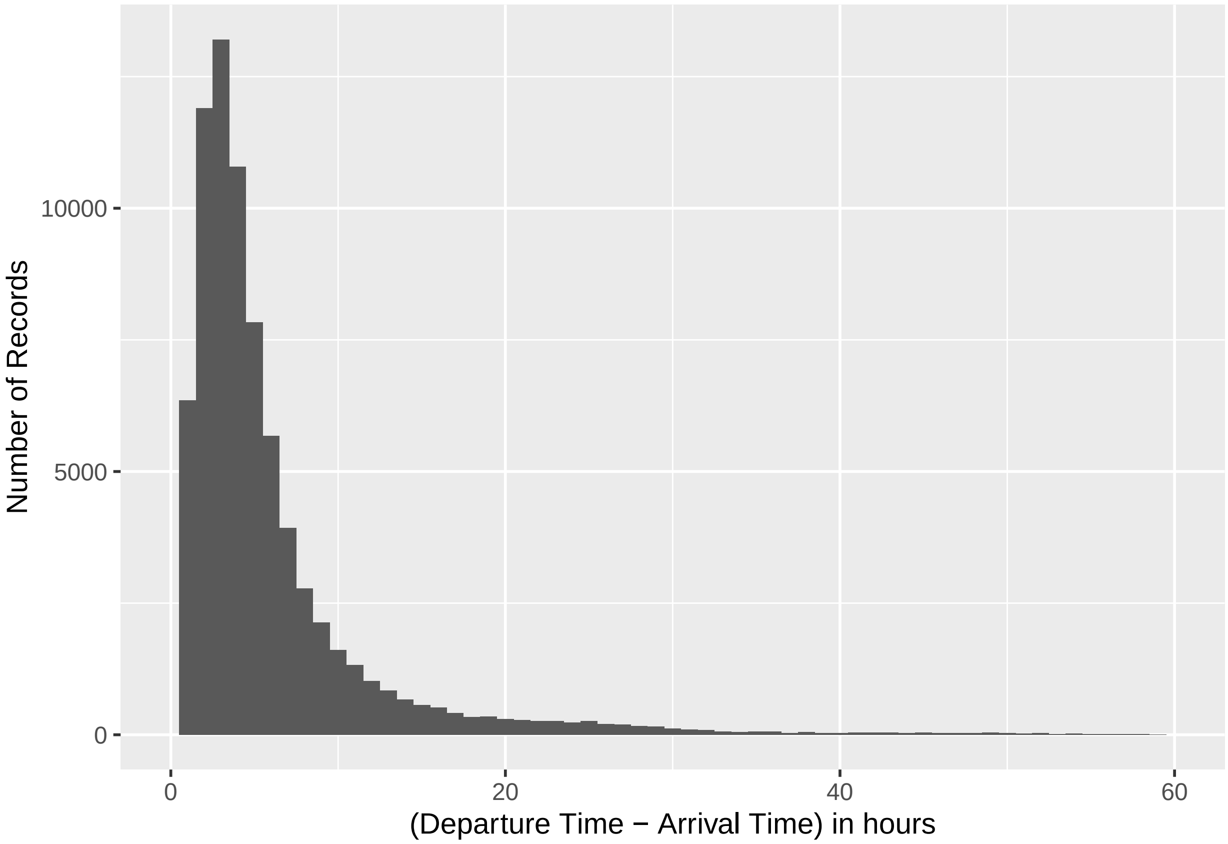
```
ggsave("los.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
ed_data_dedup %>%
    filter(startToEnd >= 0) %>%
    ggplot(aes(x = startToEnd)) + geom_histogram(binwidth = 1) + xlim(0, 60) + labs(x = "(Departure Tim
    y = "Number of Records")
```

```
## Warning: Removed 383 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



```r
ggsave("los_zoom.png")
```

```
## Saving 6.5 x 4.5 in image

## Warning: Removed 383 rows containing non-finite values (stat_bin).

## Warning: Removed 2 rows containing missing values (geom_bar).
```

```r
startToEndMean = mean(ed_data_dedup$startToEnd[ed_data_dedup$startToEnd > 0], na.rm = TRUE)
startToEndSd = sd(ed_data_dedup$startToEnd[ed_data_dedup$startToEnd > 0], na.rm = TRUE)

startToEndMean
```

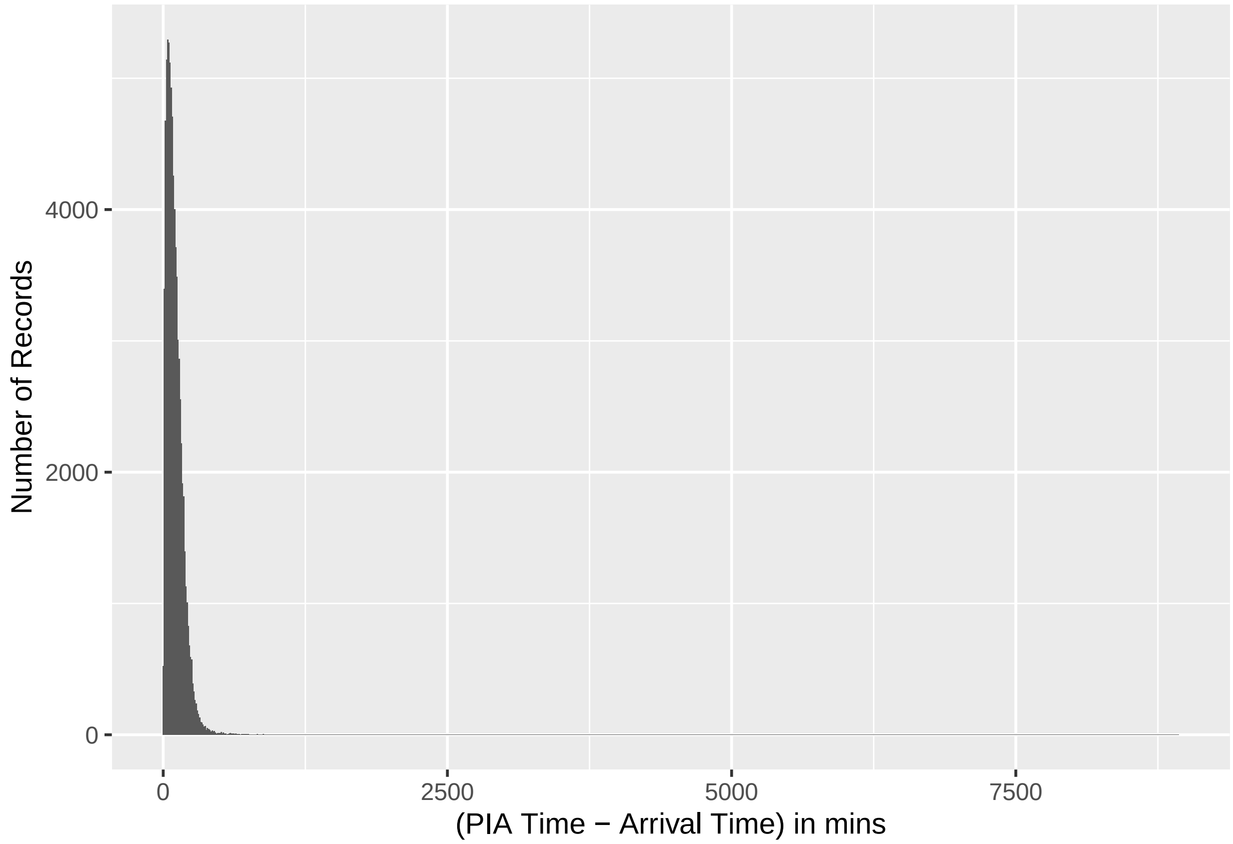
```
## [1] 6.417152
```

```r
startToEndSd
```

```
## [1] 9.136179
```

Time to seeing a physician

```r
ed_data_dedup$startToPia = as.numeric(difftime(ed_data_dedup$ed_pia_time, ed_data_dedup$ed_start_time),
    units = "mins")

ed_data_dedup %>%
    filter(ed_pia_time != ymd("2099-01-01")) %>%
    filter(startToPia > 0) %>%
    ggplot(aes(x = startToPia)) + geom_histogram(binwidth = 10) + labs(x = "(PIA Time - Arrival Time) i
    y = "Number of Records")
```



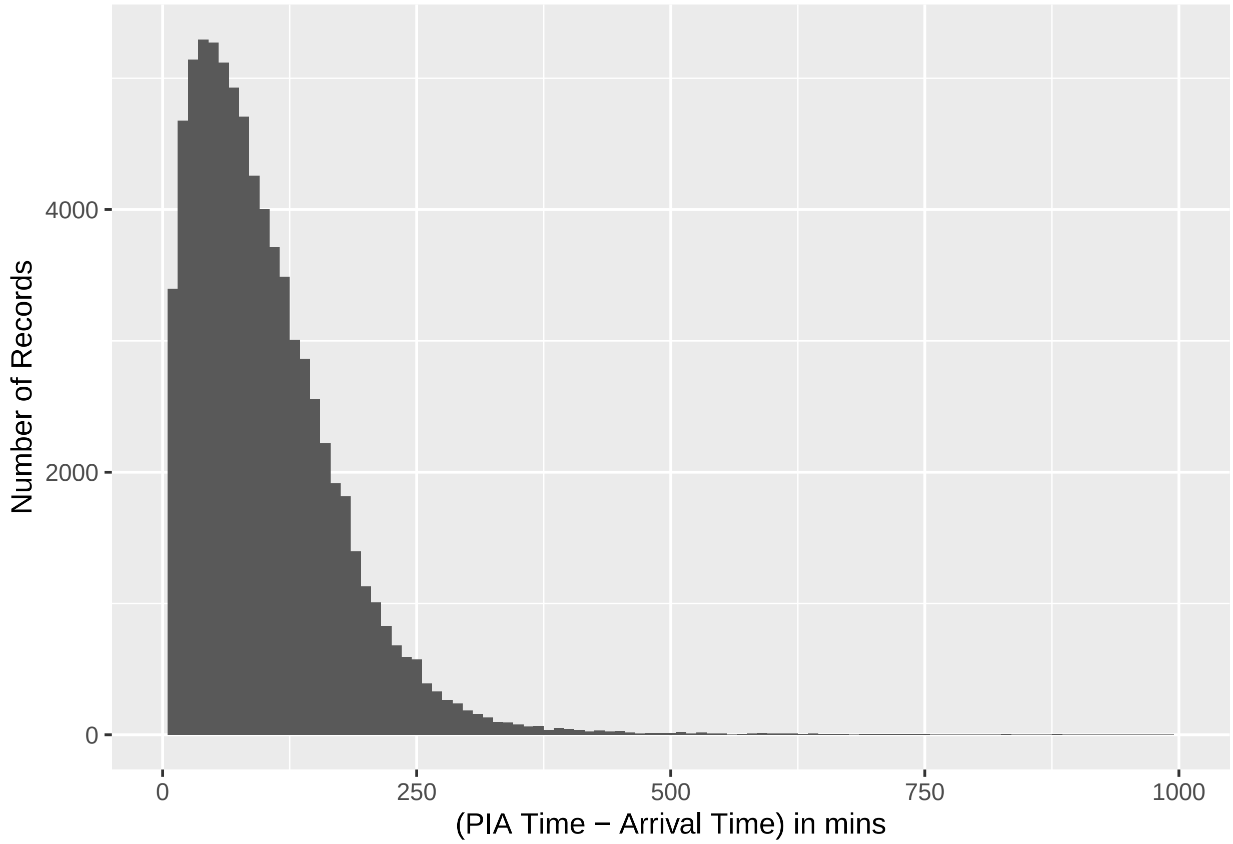
```
ggsave("pia.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
ed_data_dedup %>%
    filter(ed_pia_time != ymd("2099-01-01")) %>%
    filter(startToPia > 0) %>%
    ggplot(aes(x = startToPia)) + geom_histogram(binwidth = 10) + xlim(0, 1000) +
    labs(x = "(PIA Time - Arrival Time) in mins", y = "Number of Records")
```

```
## Warning: Removed 69 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



```
ggsave("pia_zoom.png")
```

```
## Saving 6.5 x 4.5 in image

## Warning: Removed 69 rows containing non-finite values (stat_bin).

## Warning: Removed 2 rows containing missing values (geom_bar).
```

```
startToPiaMean = mean(ed_data_dedup$startToPia[ed_data_dedup$startToPia > 0 & ed_data_dedup$ed_pia_time
    ymd("2099-01-01")], na.rm = TRUE)
startToPiaSd = sd(ed_data_dedup$startToPia[ed_data_dedup$startToPia > 0 & ed_data_dedup$ed_pia_time !=
    ymd("2099-01-01")], na.rm = TRUE)

startToPiaMean
```

```
## [1] 102.8916
```

```
startToPiaSd
```

```
## [1] 104.6482
```

Counts and proportions of presenting complaints

```
PC_counts = ed_data_dedup %>%
    count(presenting_complaint) %>%
    arrange(desc(n))

jpeg(filename = "PC.jpg", width = 800, height = 600)
pie(PC_counts$n, labels = PC_counts$presenting_complaint, radius = 0.9, cex = 1.2)
```

```
dev.off()
```

```
## pdf
##   2
```

```
PC_counts
```

```
## # A tibble: 17 x 2
##    presenting_complaint       n
##    <chr>                  <int>
##  1 abdominal pain         14470
##  2 sore throat            13708
##  3 loss of hearing         8052
##  4 confusion               7718
##  5 headache                6948
##  6 upper extremity injury  4131
##  7 lower extremity injury  4064
##  8 back pain               3600
##  9 rash                    3478
## 10 general weakness        2760
## 11 chest pain              2755
## 12 traumatic injury        2606
## 13 hallucinations          2075
## 14 bizarre behaviour       1587
## 15 burn                    1229
## 16 trouble breathing        817
## 17 unknown                  251
```

CTAS

```
CTAS_counts = ed_data_dedup %>%
    count(CTAS_DESCR) %>%
    arrange(desc(n))

jpeg(filename = "CTAS.jpg", width = 800, height = 600)
pie(CTAS_counts$n, labels = CTAS_counts$CTAS_DESCR, radius = 1, cex = 1)
dev.off()
```

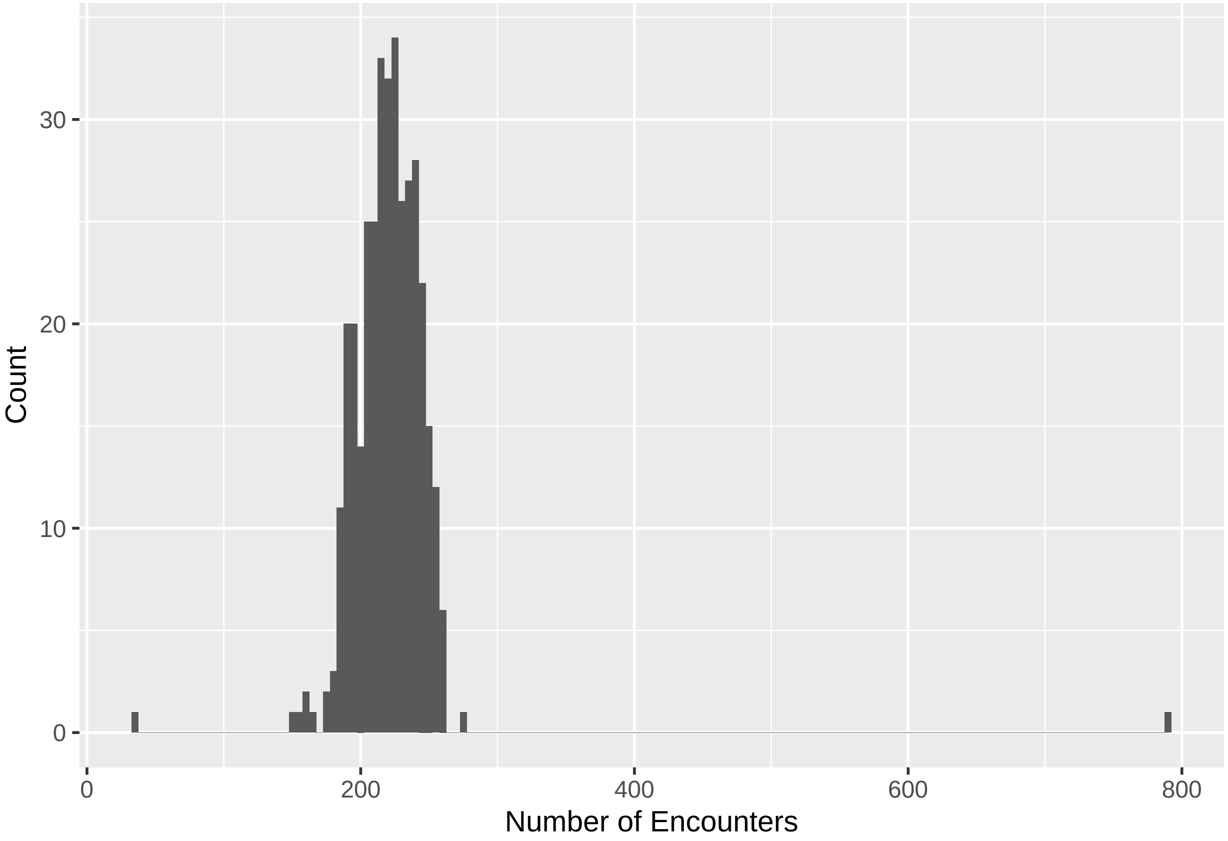
```
## pdf
##   2
```

```
CTAS_counts
```

```
## # A tibble: 6 x 2
##   CTAS_DESCR         n
##   <chr>          <int>
## 1 URGENT         34688
## 2 EMERGENCY      26029
## 3 SEMI-URGENT    12009
## 4 RESUSCITATION   3289
## 5 NON URGENT      3033
## 6 N/A             1201
```

Number of encounters

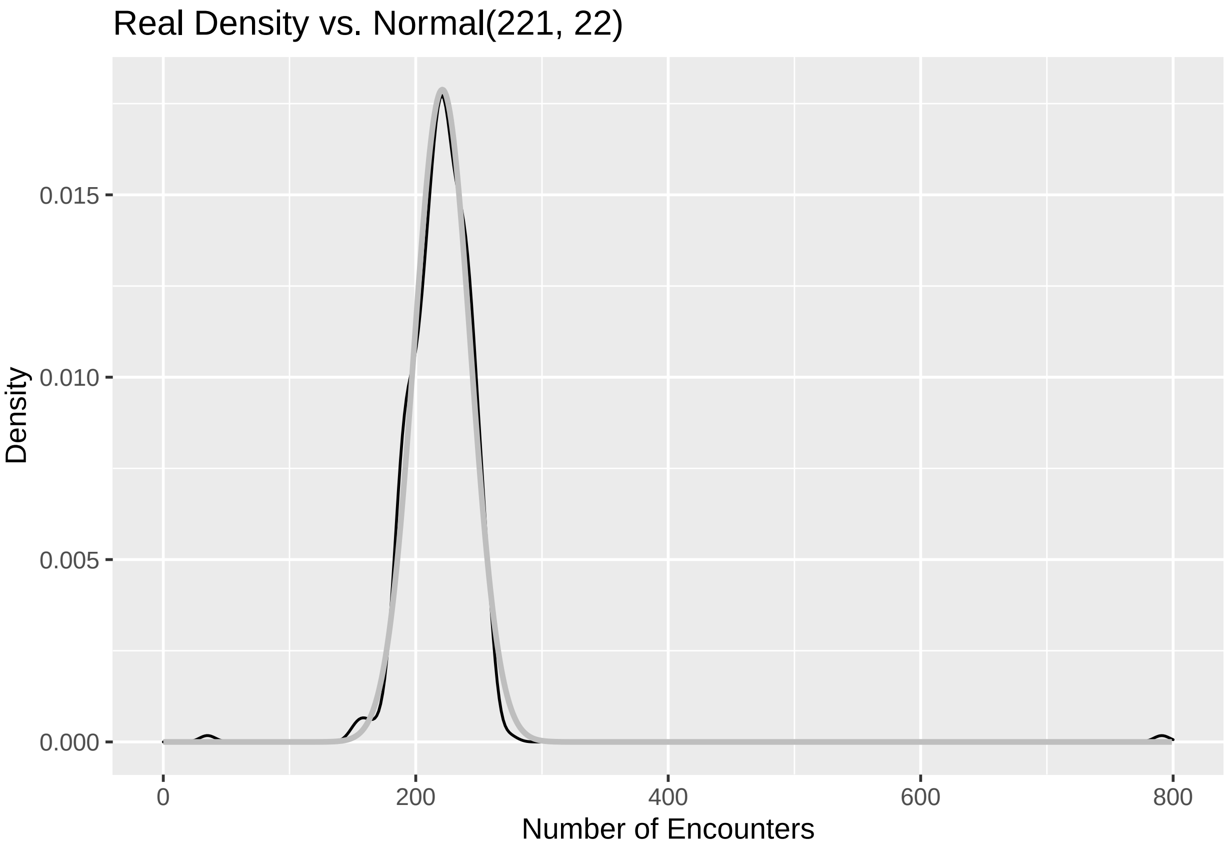
```
ed_data_dedup$ed_start_time_YMD = as_date(ed_data_dedup$ed_start_time)

n_encounters = count(ed_data_dedup, ed_start_time_YMD)

ed_data_dedup %>%
    count(ed_start_time_YMD) %>%
    ggplot(aes(x = n)) + geom_histogram(binwidth = 5) + labs(x = "Number of Encounters",
    y = "Count")
```



```
ggsave("n_enc.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
n_encountersMean = mean(n_encounters$n, rm.na=TRUE)
n_encountersSd = sd(n_encounters$n)

n_encounters %>%
  ggplot(aes(x=n)) +
  geom_density() +
  stat_function(fun=dnorm, args=c(n_encountersMean, n_encountersSd / 1.7), xlim=c(0, 799), n=800, size=
  xlim(0,800) +
  labs(x = "Number of Encounters", y="Density", title="Real Density vs. Normal(221, 22)")
```



```
ggsave("n_enc_fit.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
n_encountersMean
```

```
## [1] 221.0716
```

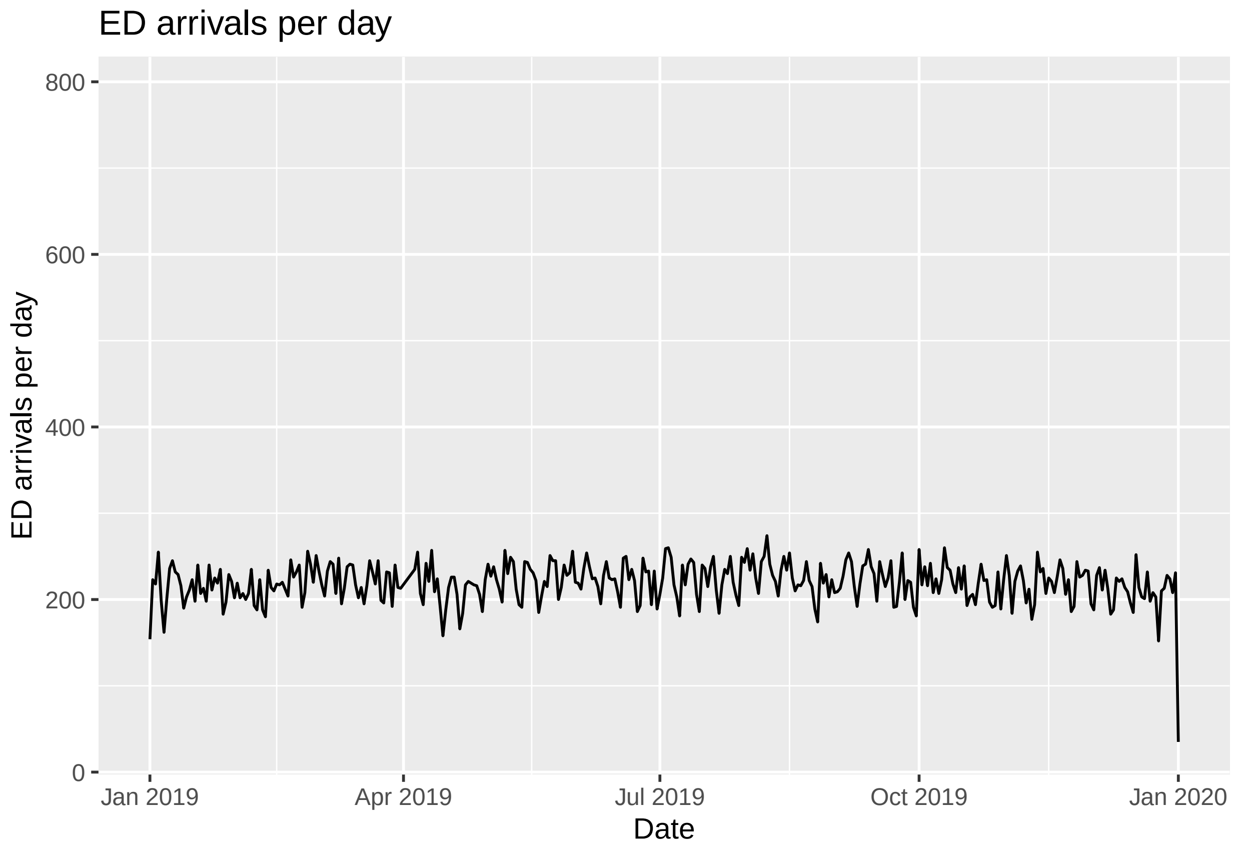
```
n_encountersSd
```

```
## [1] 37.91762
```

Time series

```
ggplot(n_encounters, aes(x = ed_start_time_YMD, y = n)) + geom_line() + labs(x = "Date",
    y = "ED arrivals per day", title = "ED arrivals per day")
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



```
ggsave("ed_arr.png")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

```
n_encountersMean
```

```
## [1] 221.0716
```

```
tail(n_encounters)
```

```
## # A tibble: 6 x 2
##   ed_start_time_YMD     n
##   <date>            <int>
## 1 2019-12-28          228
## 2 2019-12-29          224
## 3 2019-12-30          208
## 4 2019-12-31          231
## 5 2020-01-01           35
## 6 NA                  791
```

Weekend vs. weekday volumes

```
weekend_names = c("Sat", "Sun")

# Get weekday and weekend dfs
weekday_encounters = n_encounters %>%
    filter(!weekdays(ed_start_time_YMD, abbreviate = TRUE) %in% weekend_names) %>%
    filter(!is.na(ed_start_time_YMD))
weekend_encounters = n_encounters %>%
    filter(weekdays(ed_start_time_YMD, abbreviate = TRUE) %in% weekend_names)

# Get weekday stats
print("Weekday mean:")
```

```
## [1] "Weekday mean:"
```

```
mean(weekday_encounters$n)
```

```
## [1] 222.0426
```

```
print("Weekday standard deviation:")
```

```
## [1] "Weekday standard deviation:"
```

```
sd(weekday_encounters$n)
```

```
## [1] 24.28034
```

```
# Get weekend stats
print("Weekend mean:")
```

```
## [1] "Weekend mean:"
```

```
mean(weekend_encounters$n)
```

```
## [1] 213.1827
```

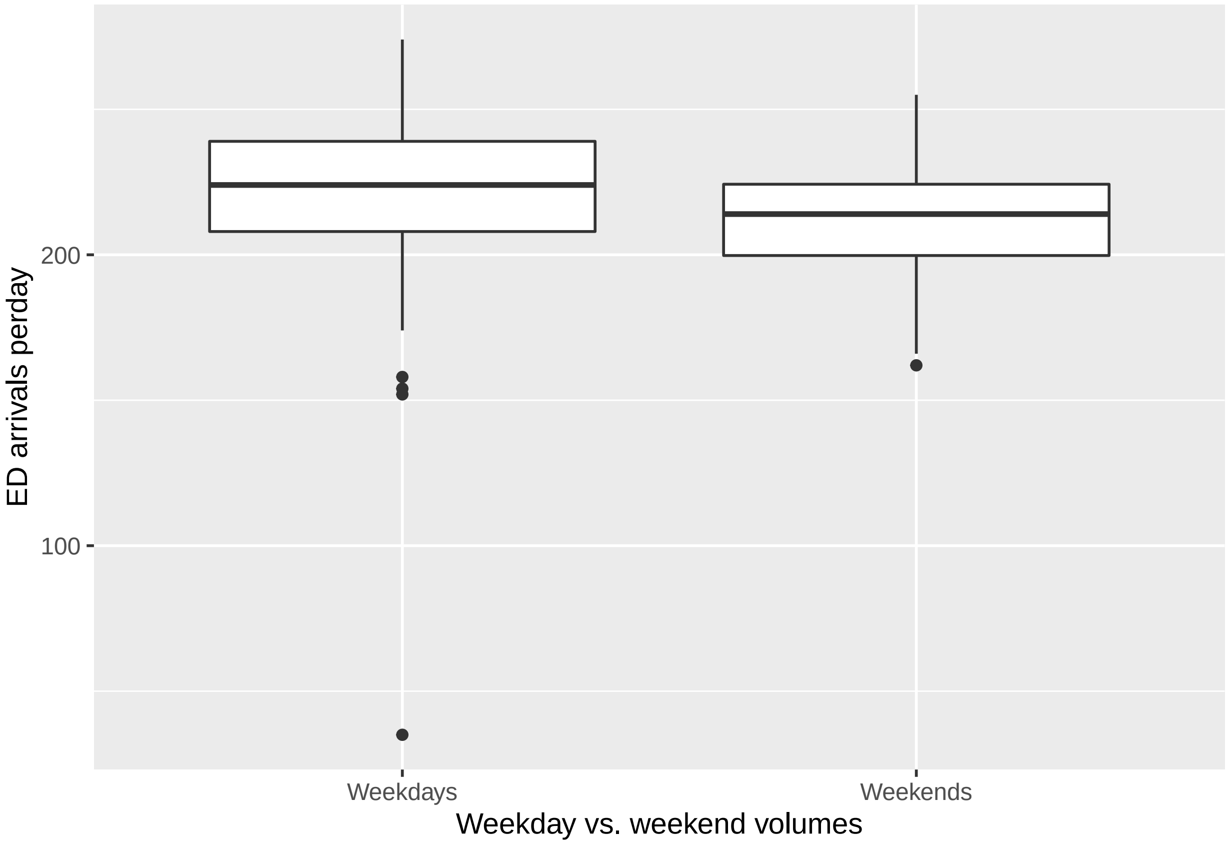
```
print("Weekend standard deviation:")
```

```
## [1] "Weekend standard deviation:"
```

```
sd(weekend_encounters$n)
```

```
## [1] 19.06034
```

```
weekday_encounters$day = "Weekdays"
weekend_encounters$day = "Weekends"
all_encounters = rbind(weekday_encounters, weekend_encounters)

all_encounters %>%
    ggplot(aes(x = day, y = n)) + geom_boxplot() + labs(x = "Weekday vs. weekend volumes",
    y = "ED arrivals perday")
```



```
ggsave("wkd_wke.png")
```

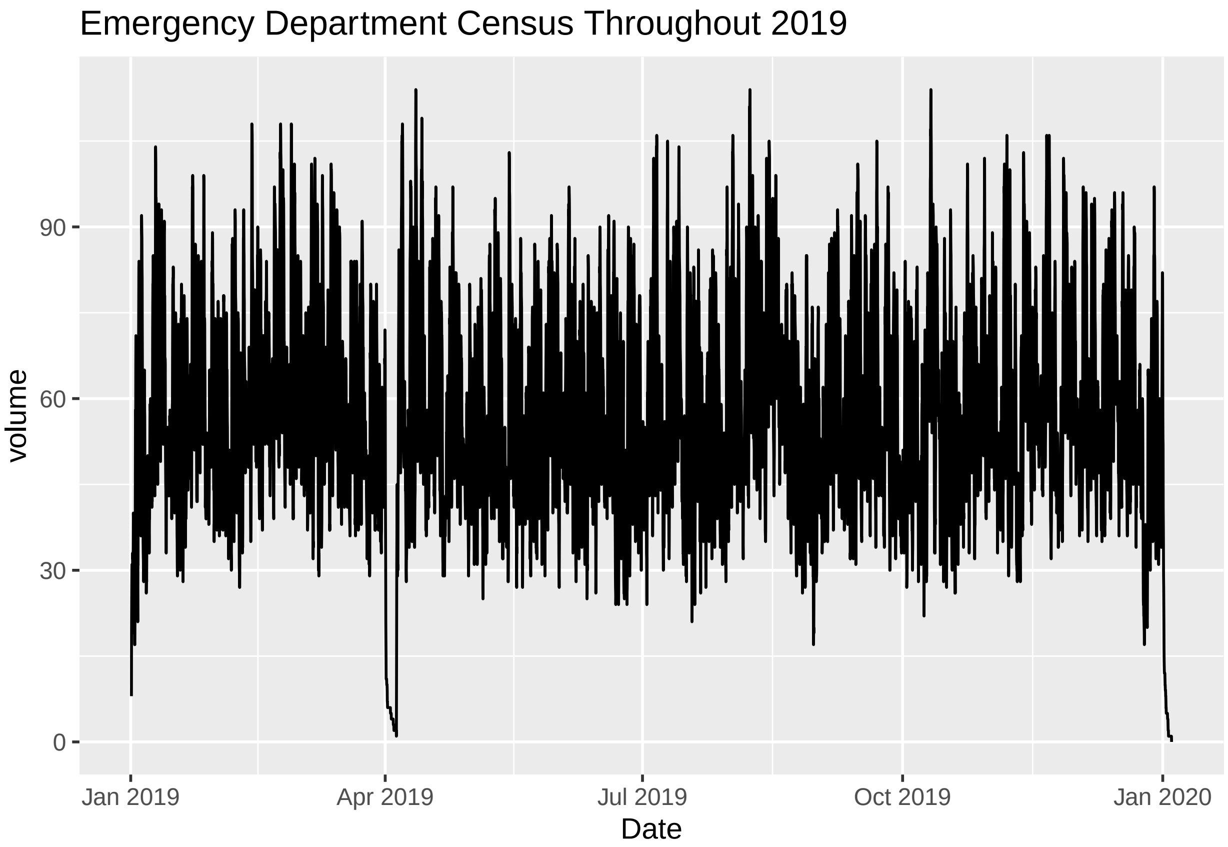
```
## Saving 6.5 x 4.5 in image
```

```
t.test(n ~ day, data = all_encounters, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  n by day
## t = 3.3297, df = 360, p-value = 0.0009593
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   3.627101 14.092786
## sample estimates:
## mean in group Weekdays mean in group Weekends
##               222.0426               213.1827
```

Working hours

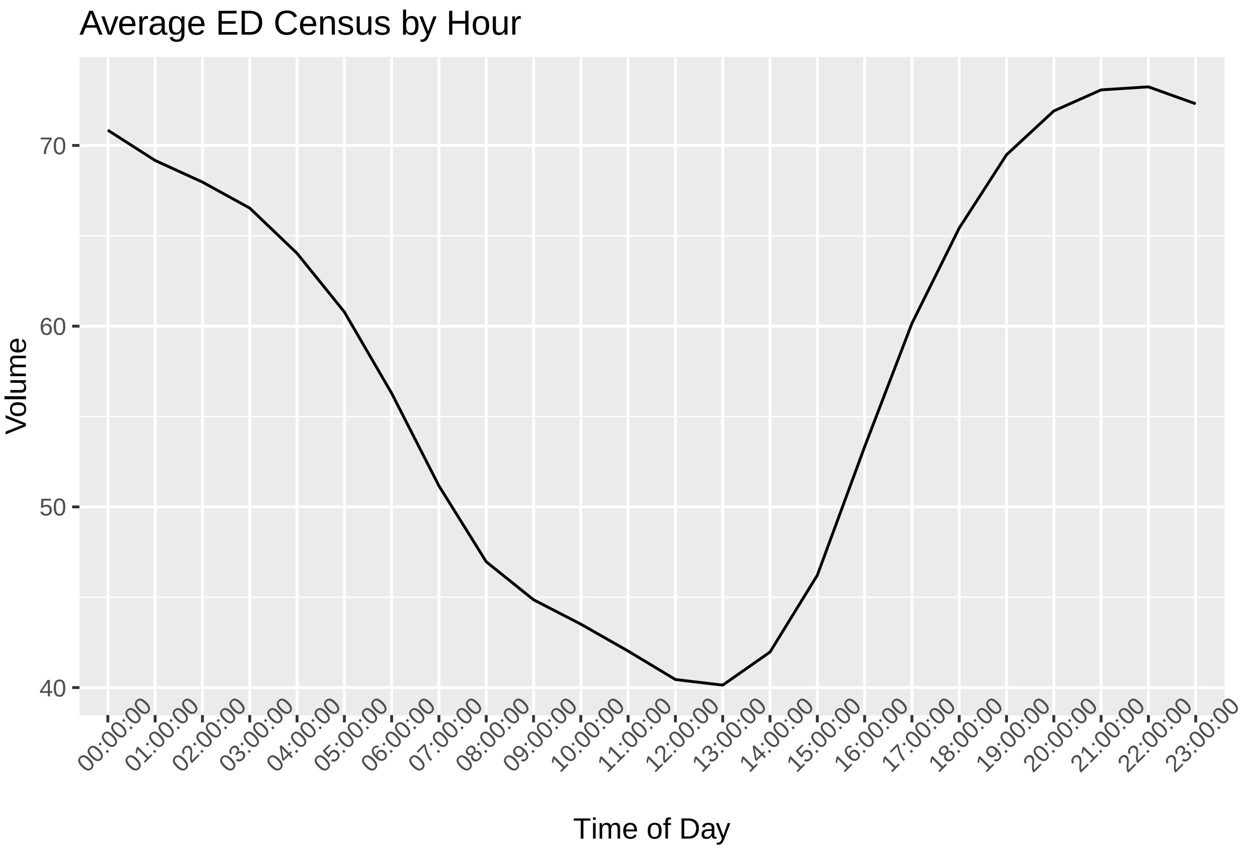
```
ed_data_clean <- ed_data_dedup %>%
    filter(!is.na(ed_start_time), !is.na(ed_end_time)) %>%
    filter(ed_start_time < ed_end_time) %>%
    mutate(ed_start_time = floor_date(ed_start_time, unit = "hour"), ed_end_time = floor_date(ed_end_ti
        unit = "hour"))

arrivals <- ed_data_clean %>%
    select(timestamp = ed_start_time) %>%
    mutate(counter = 1)

departures <- ed_data_clean %>%
    select(timestamp = ed_end_time) %>%
    mutate(counter = -1)

census_volumes <- arrivals %>%
    bind_rows(departures) %>%
    arrange(timestamp, counter) %>%
    mutate(volume = cumsum(counter))

start <- min(ed_data_clean$ed_start_time)
end <- max(ed_data_clean$ed_end_time)
full_time_window <- tibble(timestamp = seq(start, end, by = "hours"))

census_volumes <- census_volumes %>%
    right_join(full_time_window, by = "timestamp") %>%
    arrange(timestamp) %>%
    fill(volume, .direction = "down")

census_volumes <- census_volumes %>%
    arrange(timestamp, volume) %>%
    group_by(timestamp) %>%
    summarise_all(last)

census_volumes %>%
    ggplot(aes(timestamp, volume)) + geom_line() + labs(x = "Date", title = "Emergency Department Censu
```



```
ggsave("census_ydm_hms.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
census_hours <- census_volumes %>%
    mutate(timestamp = format(timestamp, format = "%T"))

aggregate(x = census_hours$volume, by = list(timestamp = census_hours$timestamp),
    FUN = mean) %>%
    ggplot(aes(timestamp, x, group = 1)) + geom_line() + labs(y = "Volume", x = "Time of Day",
    title = "Average ED Census by Hour") + theme(axis.text.x = element_text(angle = 45))
```



```
ggsave("census_hms.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
census_by_hour = aggregate(x = census_hours$volume, by =list(timestamp=census_hours$timestamp), FUN=mean

summary(census_by_hour)
```

```
##   timestamp               x
##  Length:24          Min.   :40.14
##  Class :character   1st Qu.:45.88
##  Mode  :character   Median :60.47
##                     Mean   :57.99
##                     3rd Qu.:69.25
##                     Max.   :73.24
```

Descriptive Summary

```
ed_data_dedup %>%
    filter(startToEnd >= 0) %>%
    filter(ed_pia_time != ymd("2099-01-01")) %>%
    filter(startToPia >= 0) %>%
    summary()
```

```
##  ENCOUNTER_NUM     CTAS_CD            CTAS_DESCR
##  Min.   :    1   Length:75940       Length:75940
##  1st Qu.:20930   Class :character   Class :character
##  Median :41048   Mode  :character   Mode  :character
##  Mean   :40826
##  3rd Qu.:61159
##  Max.   :81133
##
##  ed_start_time                  ed_end_time
##  Min.   :2019-01-01 06:06:00   Min.   :2019-01-01 06:43:00
##  1st Qu.:2019-04-06 05:36:30   1st Qu.:2019-04-06 16:34:45
##  Median :2019-07-05 04:24:30   Median :2019-07-05 13:19:00
##  Mean   :2019-07-03 21:40:02   Mean   :2019-07-04 04:04:32
##  3rd Qu.:2019-10-01 19:53:00   3rd Qu.:2019-10-02 01:55:45
##  Max.   :2020-01-01 05:38:00   Max.   :2020-01-04 03:08:00
##
##   ed_pia_time                  adm_start_time                   admitted
##  Min.   :2019-01-01 06:22:00   Min.   :2019-01-01 07:56:00   Min.   :0.000
##  1st Qu.:2019-04-06 07:41:45   1st Qu.:2019-03-31 22:23:15   1st Qu.:0.000
##  Median :2019-07-05 07:47:00   Median :2019-07-06 20:37:30   Median :0.000
##  Mean   :2019-07-03 23:23:02   Mean   :2019-07-04 02:20:40   Mean   :0.144
##  3rd Qu.:2019-10-01 21:54:00   3rd Qu.:2019-10-03 06:26:15   3rd Qu.:0.000
##  Max.   :2020-01-01 09:21:00   Max.   :2020-01-01 11:48:00   Max.   :1.000
##                                NA's   :65006
##       los         presenting_complaint   startToEnd        startToPia
##  Min.   : 0.100   Length:75940         Min.   :  0.100   Min.   :   3
##  1st Qu.: 2.500   Class :character     1st Qu.:  2.500   1st Qu.:  46
##  Median : 4.017   Mode  :character     Median :  4.017   Median :  85
##  Mean   : 5.780                        Mean   :  6.408   Mean   : 103
##  3rd Qu.: 6.750                        3rd Qu.:  6.750   3rd Qu.: 139
##  Max.   :24.000                        Max.   :169.550   Max.   :8935
##
##  ed_start_time_YMD
##  Min.   :2019-01-01
##  1st Qu.:2019-04-06
##  Median :2019-07-05
##  Mean   :2019-07-03
##  3rd Qu.:2019-10-01
##  Max.   :2020-01-01
##
```