Time-Date and text manipulation in R

Surya Lamichhane

24 February 2025

Contents

1. Learning Objectives

By the end of this lesson, you will be able to:

- Understand structure of time and date in R
- Manipulate time and date in R
- Manipulate strings in R

2. Time and date manupulation in R

2.1 data and time in R

```
Sys.Date()
```

```
## [1] "2025-02-24"
```

```
Sys.time()
```

```
## [1] "2025-02-24 09:17:21 CST"
```

```
Sys.timezone()
```

```
## [1] "America/Chicago"
```

2.2 lubridate package in R

The lubridate package in R is a powerful tool for working with date-time data. It simplifies parsing, manipulation, and arithmetic operations on dates and times.

```r
library(lubridate)
```

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

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

```r
now()
```

```
## [1] "2025-02-24 09:17:21 CST"
```

```r
today()
```

```
## [1] "2025-02-24"
```

2.3 Creating date/times

There are three types of date/time data that refer to an instant in time:

- date: of type
- time: of type
- date-time: of type , it uniquely identifies an instant in time

In most cases, we create a date/time in three ways:

- a string.
- an individual date-time components.
- an existing date/time object.

```r


ymd("2023-01-30")
```

Using a string

```
## [1] "2023-01-30"
```

```r
mdy("January 30, 2023")
```

```
## [1] "2023-01-30"
```

```r
mdy("January 31st, 2023")
```

```
## [1] "2023-01-31"
```

```r
dmy("30-Jan-2023")
```

```
## [1] "2023-01-30"
```

- date-time from a date by supplying a timezone:

```r
#Coordinated Universal Time (UTC)
ymd(20230131, tz = "UTC")
```

```
## [1] "2023-01-31 UTC"
```

2.4 Application (New York flight data from R package nycflights13")

```r
#install.packages("nycflights13")
library(nycflights13) #TO GET THE FILEGT DATA
library(dplyr)
str(flights)
```

```
## tibble [336,776 x 19] (S3: tbl_df/tbl/data.frame)
##  $ year          : int [1:336776] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ month         : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
##  $ dep_time      : int [1:336776] 517 533 542 544 554 554 555 557 557 558 ...
##  $ sched_dep_time: int [1:336776] 515 529 540 545 600 558 600 600 600 600 ...
##  $ dep_delay     : num [1:336776] 2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int [1:336776] 830 850 923 1004 812 740 913 709 838 753 ...
##  $ sched_arr_time: int [1:336776] 819 830 850 1022 837 728 854 723 846 745 ...
##  $ arr_delay     : num [1:336776] 11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ carrier       : chr [1:336776] "UA" "UA" "AA" "B6" ...
##  $ flight        : int [1:336776] 1545 1714 1141 725 461 1696 507 5708 79 301 ...
##  $ tailnum       : chr [1:336776] "N14228" "N24211" "N619AA" "N804JB" ...
##  $ origin        : chr [1:336776] "EWR" "LGA" "JFK" "JFK" ...
##  $ dest          : chr [1:336776] "IAH" "IAH" "MIA" "BQN" ...
##  $ air_time      : num [1:336776] 227 227 160 183 116 150 158 53 140 138 ...
##  $ distance      : num [1:336776] 1400 1416 1089 1576 762 ...
##  $ hour          : num [1:336776] 5 5 5 5 6 5 6 6 6 6 ...
##  $ minute        : num [1:336776] 15 29 40 45 0 58 0 0 0 0 ...
##  $ time_hour     : POSIXct[1:336776], format: "2013-01-01 05:00:00" "2013-01-01 05:00:00" ...
```

```r
flights %>%
  select(year, month, day, hour, minute) %>%
  head()
```

```
## # A tibble: 6 x 5
##    year month   day  hour minute
##   <int> <int> <int> <dbl>  <dbl>
## 1  2013     1     1     5     15
## 2  2013     1     1     5     29
## 3  2013     1     1     5     40
## 4  2013     1     1     5     45
## 5  2013     1     1     6      0
## 6  2013     1     1     5     58
```

```
library(lubridate)
```

```
flights %>%
  select(year, month, day, hour, minute) %>%
  mutate(departure = make_datetime(year, month, day, hour, minute))%>%
  head()
```

```
## # A tibble: 6 x 6
##    year month   day  hour minute departure
##   <int> <int> <int> <dbl>  <dbl> <dttm>
## 1  2013     1     1     5     15 2013-01-01 05:15:00
## 2  2013     1     1     5     29 2013-01-01 05:29:00
## 3  2013     1     1     5     40 2013-01-01 05:40:00
## 4  2013     1     1     5     45 2013-01-01 05:45:00
## 5  2013     1     1     6      0 2013-01-01 06:00:00
## 6  2013     1     1     5     58 2013-01-01 05:58:00
```

```
subset_flight = flights %>% select(dep_time, sched_dep_time, arr_time, sched_arr_time)
head(subset_flight)
```

```
## # A tibble: 6 x 4
##   dep_time sched_dep_time arr_time sched_arr_time
##      <int>          <int>    <int>          <int>
## 1      517            515      830            819
## 2      533            529      850            830
## 3      542            540      923            850
## 4      544            545     1004           1022
## 5      554            600      812            837
## 6      554            558      740            728
```

```
make_datetime_60 <- function(year, month, day, time) {
  make_datetime(year, month, day, time %/% 60, time %% 60)
}

flights_dt <- flights %>%
  filter(!is.na(dep_time), !is.na(arr_time)) %>%
  mutate(
    dep_time = make_datetime_60(year, month, day, dep_time),
    arr_time = make_datetime_60(year, month, day, arr_time),
    sched_dep_time = make_datetime_60(year, month, day, sched_dep_time),
    sched_arr_time = make_datetime_60(year, month, day, sched_arr_time)
  ) %>%
  select(origin, dest, ends_with("delay"), ends_with("time"))

head(flights_dt)
```

```
## # A tibble: 6 x 9
##   origin dest  dep_delay arr_delay dep_time            sched_dep_time
##   <chr>  <chr>     <dbl>     <dbl> <dttm>              <dttm>
## 1 EWR    IAH           2        11 2013-01-01 08:37:00 2013-01-01 08:35:00
## 2 LGA    IAH           4        20 2013-01-01 08:53:00 2013-01-01 08:49:00
## 3 JFK    MIA           2        33 2013-01-01 09:02:00 2013-01-01 09:00:00
## 4 JFK    BQN          -1       -18 2013-01-01 09:04:00 2013-01-01 09:05:00
## 5 LGA    ATL          -6       -25 2013-01-01 09:14:00 2013-01-01 10:00:00
## 6 EWR    ORD          -4        12 2013-01-01 09:14:00 2013-01-01 09:18:00
## # i 3 more variables: arr_time <dttm>, sched_arr_time <dttm>, air_time <dbl>
```

```r
flights_dt %>%
  select(origin,dest,arr_time,sched_arr_time,dep_time,sched_dep_time ) %>%
  mutate(arr_delay1 = (arr_time - sched_arr_time), dep_delay1 = (dep_time - sched_dep_time)) %>%
  select(origin,dest,arr_delay1,dep_delay1)  %>%
  head()
```

2.4.0 Find arrival delay and departure delay

```
## # A tibble: 6 x 4
##   origin dest  arr_delay1 dep_delay1
##   <chr>  <chr> <drtn>     <drtn>
## 1 EWR    IAH     660 secs   120 secs
## 2 LGA    IAH    1200 secs   240 secs
## 3 JFK    MIA    4380 secs   120 secs
## 4 JFK    BQN   -1080 secs   -60 secs
## 5 LGA    ATL   -1500 secs -2760 secs
## 6 EWR    ORD     720 secs  -240 secs
```

2.4.1 Exercise

Compute the number of flight that are arrived late at least 30 minutes

```r
flights_dt %>%
  select(origin,dest,arr_delay) %>%
  filter(arr_delay > 30) %>%
  head()
```

```
## # A tibble: 6 x 3
##   origin dest  arr_delay
##   <chr>  <chr>     <dbl>
## 1 JFK    MIA          33
## 2 LGA    DFW          31
## 3 EWR    ORD          32
## 4 LGA    DFW          48
## 5 JFK    LAX          44
## 6 LGA    DFW          31
```

2.4.2 Exercise

What origin has most arrival delay?

```
flights_dt %>%
  select(origin,arr_delay) %>%
  group_by(origin) %>%
  arrange(desc(arr_delay)) %>%
  head()
```

```
## # A tibble: 6 x 2
## # Groups:   origin [2]
##   origin arr_delay
##   <chr>      <dbl>
## 1 JFK         1272
## 2 JFK         1127
## 3 EWR         1109
## 4 JFK         1007
## 5 JFK          989
## 6 JFK          931
```

2.4.3 Exercise Compute the the destinations that have most departure delays.

```
flights_dt %>%
  select(dest,dep_delay) %>%
  group_by(dest) %>%
  arrange(desc(dep_delay))%>%
  head(10)
```

```
## # A tibble: 10 x 2
## # Groups:   dest [10]
##    dest  dep_delay
##    <chr>     <dbl>
##  1 HNL        1301
##  2 CMH        1137
##  3 ORD        1126
##  4 SFO        1014
##  5 CVG        1005
##  6 TPA         960
##  7 MSP         911
##  8 PDX         899
##  9 ATL         898
## 10 MIA         896
```

3. String Manipulation in R using *stringr* package

```
string <- "Hi, how are you doing?"
vector_of_strings <- c("Hi, how are you doing?", "I'm doing well, HBY?", "Me too, thanks for asking.")
```

```
cat("number of characters in string:", nchar(string), "\n")
```

```
## number of characters in string: 22
```

```
cat("number of characters in vector_of_strings:", nchar(vector_of_strings))
```

```
## number of characters in vector_of_strings: 22 20 26
```

3.1 Practices on basic functions in stringr

```
library(stringr)
str_to_lower(vector_of_strings)
```

```
## [1] "hi, how are you doing?"     "i'm doing well, hby?"
## [3] "me too, thanks for asking."
```

```
str_to_upper(vector_of_strings)
```

```
## [1] "HI, HOW ARE YOU DOING?"     "I'M DOING WELL, HBY?"
## [3] "ME TOO, THANKS FOR ASKING."
```

```
str_to_title(vector_of_strings)
```

```
## [1] "Hi, How Are You Doing?"     "I'm Doing Well, Hby?"
## [3] "Me Too, Thanks For Asking."
```

```
str_to_sentence(vector_of_strings)
```

```
## [1] "Hi, how are you doing?"     "I'm doing well, hby?"
## [3] "Me too, thanks for asking."
```

```
str_length(vector_of_strings)
```

```
## [1] 22 20 26
```

```


str_sub(vector_of_strings, start = 1, end =4) # extracting first to fifth character
```

3.1 Extracting particular characters

```
## [1] "Hi, " "I'm " "Me t"
```

```
str_sub(vector_of_strings, start = -5, end = -1) # extracting fifth-to-last to last character
```

```
## [1] "oing?" " HBY?" "king."
```

```
str_sub(vector_of_strings, start = -1) <- "!"
vector_of_strings
```

3.2. Add a character

```
## [1] "Hi, how are you doing!"     "I'm doing well, HBY!"
## [3] "Me too, thanks for asking!"
```

```
paste(vector_of_strings , "!", sep = '')
```

```
## [1] "Hi, how are you doing!!"     "I'm doing well, HBY!!"
## [3] "Me too, thanks for asking!!"
```

```
vector_of_strings
```

```
## [1] "Hi, how are you doing!"     "I'm doing well, HBY!"
## [3] "Me too, thanks for asking!"
```

```
unnecessary_whitespaces <- c("    on the left", "on the right    ", "    on both sides   ", "   literall
unnecessary_whitespaces
```

3.3 Removing unnecessary whitespaces

```
## [1] "    on the left"              "on the right    "
## [3] "    on both sides   "         "   literally    everywhere  "
```

```
str_trim(unnecessary_whitespaces, side = "left")
```

```
## [1] "on the left"               "on the right    "
## [3] "on both sides   "          "literally    everywhere  "
```

```
str_trim(unnecessary_whitespaces, side = "both") # the default option
```

```
## [1] "on the left"             "on the right"
## [3] "on both sides"           "literally    everywhere"
```

Not fixing at middle

```
str_squish(unnecessary_whitespaces)
```

```
## [1] "on the left"          "on the right"         "on both sides"
## [4] "literally everywhere"
```

3.4 Extract certain words or values based on what comes before or after them

- List all the numbers that appear before “m.”

```
heights <- c("1m30cm", "2m01cm", "3m10cm")
str_view(heights, "[0-9]+(?=m)")
```

```
## [1] | <1>m30cm
## [2] | <2>m01cm
## [3] | <3>m10cm
```

- List all the numbers that appear after "m."

```
str_view(heights, "[0-9]+(?!m)") # after m
```

```
## [1] | 1m<30>cm
## [2] | 2m<01>cm
## [3] | 3m<10>cm
```

3.5 Patterns Let's look at sentences dataset.

- pattern to be present in the beginning ^ or at the end $ of a string.

```
first_10_sentences = sentences[1:10]
first_10_sentences
```

```
##  [1] "The birch canoe slid on the smooth planks."
##  [2] "Glue the sheet to the dark blue background."
##  [3] "It's easy to tell the depth of a well."
##  [4] "These days a chicken leg is a rare dish."
##  [5] "Rice is often served in round bowls."
##  [6] "The juice of lemons makes fine punch."
##  [7] "The box was thrown beside the parked truck."
##  [8] "The hogs were fed chopped corn and garbage."
##  [9] "Four hours of steady work faced us."
## [10] "A large size in stockings is hard to sell."
```

```
str_view(first_10_sentences, "^The")
```

```
## [1] | <The> birch canoe slid on the smooth planks.
## [4] | <The>se days a chicken leg is a rare dish.
## [6] | <The> juice of lemons makes fine punch.
## [7] | <The> box was thrown beside the parked truck.
## [8] | <The> hogs were fed chopped corn and garbage.
```

- Find the sentences start with the and end with full stop (.)

```
temp_string = str_detect(first_10_sentences, "^The.+\\.$")
temp_string
```

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

```
str_view(first_10_sentences, "^The.+\\.$")
```

```
## [1] | <The birch canoe slid on the smooth planks.>
## [4] | <These days a chicken leg is a rare dish.>
## [6] | <The juice of lemons makes fine punch.>
## [7] | <The box was thrown beside the parked truck.>
## [8] | <The hogs were fed chopped corn and garbage.>
```

```


str_replace(heights, "m", "meters")
```

3.6 Replace string

```
## [1] "1meters30cm" "2meters01cm" "3meters10cm"
```

3.2 Practices on More advanced string manipulation

3.2.1 Detect matches

```
first_10_sentences
```

```
##  [1] "The birch canoe slid on the smooth planks."
##  [2] "Glue the sheet to the dark blue background."
##  [3] "It's easy to tell the depth of a well."
##  [4] "These days a chicken leg is a rare dish."
##  [5] "Rice is often served in round bowls."
##  [6] "The juice of lemons makes fine punch."
##  [7] "The box was thrown beside the parked truck."
##  [8] "The hogs were fed chopped corn and garbage."
##  [9] "Four hours of steady work faced us."
## [10] "A large size in stockings is hard to sell."
```

- Let's count number of 'the' in a document
- Let's try using boundary, '\b…\b' try to find the exact word between the boundries.

```
sum(str_count(str_to_lower(first_10_sentences), regex("\\bthe\\b")))
```

```
## [1] 9
```

- Let's find how many sentences has the word 'the'

```
sum(str_detect(str_to_lower(first_10_sentences), regex("\\bthe\\b")))
```

```
## [1] 6
```

For more details see **stringr manual** [https://cran.r-project.org/web/packages/stringr/stringr.pdf].

4. Application on dataset of Boston 311 Service request in 2023

```r
Boston311_2025_data <- read.csv('~/Desktop/STAT4101L_all_files/4101L-Fall-2023/Course Materials/Section

str(Boston311_2025_data)
```

```
## 'data.frame':    25021 obs. of  30 variables:
##  $ case_enquiry_id               : num  1.01e+11 1.01e+11 1.01e+11 1.01e+11 1.01e+11 ...
##  $ open_dt                       : chr  "2025-02-06 07:25:00" "2025-02-06 07:32:00" "2025-02-06 11:5
##  $ sla_target_dt                 : chr  "2025-02-07 07:25:52" "2025-03-08 07:32:02" "2025-02-07 11:5
##  $ closed_dt                     : chr  "" "" "2025-02-06 13:14:36" "2025-02-06 21:46:43" ...
##  $ on_time                       : chr  "OVERDUE" "ONTIME" "ONTIME" "ONTIME" ...
##  $ case_status                   : chr  "Open" "Open" "Closed" "Closed" ...
##  $ closure_reason                : chr  " " " " "Case Closed. Closed date : Thu Feb 06 18:14:36 EST
##  $ case_title                    : chr  "Traffic Signal Inspection" "New Sign  Crosswalk or Pavement
##  $ subject                       : chr  "Transportation - Traffic Division" "Transportation - Traffi
##  $ reason                        : chr  "Signs & Signals" "Signs & Signals" "Highway Maintenance" "E
##  $ type                          : chr  "Traffic Signal Inspection" "New Sign  Crosswalk or Pavement
##  $ queue                         : chr  "BTDT_Engineering_Request for Traffic Signal Study or Review
##  $ department                    : chr  "BTDT" "BTDT" "PWDx" "BTDT" ...
##  $ submitted_photo               : logi  NA NA NA NA NA NA ...
##  $ closed_photo                  : chr  "" "" "" "" ...
##  $ location                      : chr  "269 Forest Hills St  Jamaica Plain  MA  02130" "100 City Hal
##  $ fire_district                 : int  9 3 12 9 11 4 7 4 11 11 ...
##  $ pwd_district                  : chr  "02" "1B" "08" "10B" ...
##  $ city_council_district         : int  6 1 5 9 8 8 7 8 9 9 ...
##  $ police_district               : chr  "E13" "A1" "E18" "B2" ...
##  $ neighborhood                  : chr  "Jamaica Plain" "Downtown / Financial District" "Hyde Park"
##  $ neighborhood_services_district: int  11 3 10 13 15 14 13 14 15 15 ...
##  $ ward                          : chr  "Ward 11" "Ward 3" "18" "12" ...
##  $ precinct                      : chr  "1108" "0306" "1813" "1207" ...
##  $ location_street_name          : chr  "269 Forest Hills St" "100 City Hall Plz" "35 Harvard Ave" "
##  $ location_zipcode              : int  2130 2108 2136 NA NA 2116 2118 2115 2135 2134 ...
##  $ latitude                      : num  42.3 42.4 42.3 42.3 42.4 ...
##  $ longitude                     : num  -71.1 -71.1 -71.1 -71.1 -71.1 ...
##  $ geom_4326                     : chr  "0101000020E610000099BC27C4A8C651C00AD8068ED0264540" "010100
##  $ source                        : chr  "Citizens Connect App" "Self Service" "Citizens Connect App"
```

```r
library(dplyr)
cat("Number of recorded complaints are:",length(Boston311_2025_data$case_title))
```

4.1. How many recorded complaints are in the data set?

```
## Number of recorded complaints are: 25021
```

```r
library(stringr)
head(Boston311_2025_data$case_title)
```

4.2. What fraction of calls in this data set deal with traffic? A complaint is considered to deal with traffic if it has the word "traffic" present in the value of the "case_title" column?

```
## [1] "Traffic Signal Inspection"
## [2] "New Sign  Crosswalk or Pavement Marking"
## [3] "Empty Litter Basket"
## [4] "Parking Enforcement"
## [5] "Traffic Signal Inspection"
## [6] "Transportation General Request"
```

```r
traffic_related_complaints_status <- str_detect(str_to_lower(Boston311_2025_data$case_title), regex("\\
print(sum(traffic_related_complaints_status))
```

```
## [1] 569
```

```r
cat("proprotion of traffic related complains:", sum(traffic_related_complaints_status)/length(Boston311
```

```
## proprotion of traffic related complains: 0.0227409
```

```r
Boston311_2025_data$Parking_Enforcement_status <- str_detect(Boston311_2025_data$case_title, regex("\\b
```

```r
library(dplyr)
Parking_Enforcement_by_nbd <- Boston311_2025_data %>%
  group_by(neighborhood) %>%
  summarise(nbd_count_Parking_Enforcement = sum(Parking_Enforcement_status)) %>%
  arrange(desc(nbd_count_Parking_Enforcement))
head(Parking_Enforcement_by_nbd, 10)
```

4.3. Which neighborhood has maximum number of complaints of type "Parking Enforcement"?

```
## # A tibble: 10 x 2
##    neighborhood                             nbd_count_Parking_Enforcement
##    <chr>                                                            <int>
##  1 South Boston / South Boston Waterfront                            1126
##  2 Dorchester                                                         717
##  3 Allston / Brighton                                                 538
##  4 East Boston                                                        515
##  5 South End                                                          387
##  6 Boston                                                             368
##  7 Roxbury                                                            363
##  8 Downtown / Financial District                                      347
##  9 Jamaica Plain                                                      248
## 10 Charlestown                                                        225
```

4.4. Each call is assigned a police district. The date of the call is in column "open_dt" and the assigned police district is in column "police_district".

a. Compute how many calls are assigned to each police district every month.
b. Compute how many calls are assigned to each police district on average.
c. Compute how many calls are assigned to each month on average.
d. What is the median of this monthly number of calls per police district?

- Let's find unique police districts in the dataset

```
unique(Boston311_2025_data$police_district)
```

```
##  [1] "E13" "A1"  "E18" "B2"  "D14" "D4"  "C6"  "C11" "E5"  "A7"  "A15" "B3" 
## [13] " "   ""
```

- Let's check the open date using variable 'open_dt'

```
class(Boston311_2025_data$open_dt)
```

```
## [1] "character"
```

```
head(Boston311_2025_data$open_dt)
```

```
## [1] "2025-02-06 07:25:00" "2025-02-06 07:32:00" "2025-02-06 11:50:27"
## [4] "2025-02-06 12:20:00" "2025-02-06 12:21:00" "2025-02-06 12:45:00"
```

```
library(lubridate)
class(ymd_hms(Boston311_2025_data$open_dt))
```

```
## [1] "POSIXct" "POSIXt"
```

```
head(ymd_hms(Boston311_2025_data$open_dt))
```

```
## [1] "2025-02-06 07:25:00 UTC" "2025-02-06 07:32:00 UTC"
## [3] "2025-02-06 11:50:27 UTC" "2025-02-06 12:20:00 UTC"
## [5] "2025-02-06 12:21:00 UTC" "2025-02-06 12:45:00 UTC"
```

- Number of calls assigned to each police district by month

```
Boston311_2025_data$month <- month(ymd_hms(Boston311_2025_data$open_dt))
data_by_month_by_pd = Boston311_2025_data %>%
  group_by(month, police_district ) %>%
  summarise(count_by_month_pd = n())
```

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

```r
head(data_by_month_by_pd)
```

```
## # A tibble: 6 x 3
## # Groups:   month [1]
##   month police_district count_by_month_pd
##   <dbl> <chr>                       <int>
## 1     1 ""                              4
## 2     1 " "                           190
## 3     1 "A1"                         1960
## 4     1 "A15"                         640
## 5     1 "A7"                         1450
## 6     1 "B2"                         1646
```

- Average number of calls to each police district

```r
data_by_month_by_pd %>%
  group_by(police_district) %>%
  summarise(average_monthly_calls = mean(count_by_month_pd)) %>%
  arrange(desc(average_monthly_calls))
```

```
## # A tibble: 14 x 2
##    police_district average_monthly_calls
##    <chr>                           <dbl>
##  1 "D4"                            1938.
##  2 "C6"                            1482.
##  3 "D14"                           1348
##  4 "A1"                            1274.
##  5 "C11"                           1221
##  6 "B2"                            1032.
##  7 "A7"                             930.
##  8 "E13"                            818
##  9 "E5"                             716.
## 10 "B3"                             670.
## 11 "E18"                            549
## 12 "A15"                            402
## 13 " "                              129
## 14 ""                                 4
```

- Average number of calls by month

```r
data_by_month_by_pd %>%
  group_by(month) %>%
  summarise(average_monthly_calls = mean(count_by_month_pd))
```

```
## # A tibble: 2 x 2
##   month average_monthly_calls
##   <dbl>                 <dbl>
## 1     1                 1396.
## 2     2                  421.
```

- Median number of calls by police district

```
data_by_month_by_pd %>%
  group_by(police_district) %>%
  summarise(median_monthly_calls = median(count_by_month_pd))%>%
  arrange(desc(median_monthly_calls))
```

```
## # A tibble: 14 x 2
##    police_district median_monthly_calls
##    <chr>                          <dbl>
##  1 "D4"                           1938.
##  2 "C6"                           1482.
##  3 "D14"                          1348
##  4 "A1"                           1274.
##  5 "C11"                          1221
##  6 "B2"                           1032.
##  7 "A7"                            930.
##  8 "E13"                           818
##  9 "E5"                            716.
## 10 "B3"                            670.
## 11 "E18"                           549
## 12 "A15"                           402
## 13 " "                             129
## 14 ""                                4
```