Citibike 2023

Jana Choe

2026-02-13

Note that we couldn't upload the entire dataset as the entire zip file is 1.2 GB; instead we are uploading parts of the original dataset and will also include the knitted version for the grader to see what the code looks like

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
```

Here's an example of data preprocessing using january:

```
jan_1 <- read.csv(file = "~/Downloads/2023-citibike-tripdata/202301-citibike-tripdata_1.csv", header = T
head(jan_1, 20) #before preprocessing the data
```

```
##             ride_id rideable_type              started_at
## 1  A8518A6C4BE513DE  classic_bike 2023-01-03 23:14:52.325
## 2  A3911E4F5B9B5773 electric_bike 2023-01-07 07:57:40.054
## 3  AE7F74C32AEBF6F2 electric_bike 2023-01-09 18:37:44.830
## 4  6E10997509D2B7F6 electric_bike 2023-01-05 19:06:15.350
## 5  AA546E74A9330BD4 electric_bike 2023-01-02 20:25:23.300
## 6  961077365AFFC8CC  classic_bike 2023-01-07 18:02:43.067
## 7  6FFCB761CD78DC81 electric_bike 2023-01-04 00:36:32.861
## 8  AC167050A0CA94F9  classic_bike 2023-01-10 21:32:25.868
## 9  B531AB8CC8B061ED  classic_bike 2023-01-09 13:26:01.440
## 10 1C929CE614CF1CE8  classic_bike 2023-01-13 19:53:17.048
## 11 13E216A8F52C8FF8  classic_bike 2023-01-13 18:28:52.463
## 12 980CC9886FB35020 electric_bike 2023-01-13 17:48:16.624
## 13 1D0034F4410D09AA electric_bike 2023-01-10 10:04:44.883
## 14 0548A0D317B4989D  classic_bike 2023-01-08 11:48:57.817
## 15 D2787C8881EC8525  classic_bike 2023-01-08 03:05:29.667
```

```
## 16 A71E8A699E514B68  classic_bike 2023-01-02 15:39:55.042
## 17 BB713E7E7683F394 electric_bike 2023-01-08 19:00:02.421
## 18 116795B3505E344A electric_bike 2023-01-03 19:03:16.656
## 19 E6360387B6BAAA64 electric_bike 2023-01-11 15:45:47.446
## 20 885F90DC00AF73A5 electric_bike 2023-01-05 22:09:04.228
##                  ended_at            start_station_name start_station_id
## 1  2023-01-03 23:33:42.737               E 1 St & Bowery          5636.13
## 2  2023-01-07 08:01:27.330               E 1 St & Bowery          5636.13
## 3  2023-01-09 18:48:56.233               1 Ave & E 39 St          6303.01
## 4  2023-01-05 19:08:33.547     E Burnside Ave & Ryer Ave          8397.02
## 5  2023-01-03 10:51:25.164       Clermont Ave & Park Ave          4692.01
## 6  2023-01-07 18:04:23.030               E 14 St & 1 Ave           5779.1
## 7  2023-01-04 00:45:03.693          FDR Drive & E 35 St          6230.04
## 8  2023-01-10 21:36:46.575       Dock 72 Way & Market St          4804.02
## 9  2023-01-09 13:38:05.531       Dock 72 Way & Market St          4804.02
## 10 2023-01-13 19:59:38.873         Forsyth St & Grand St          5382.07
## 11 2023-01-13 18:46:55.073       Dock 72 Way & Market St          4804.02
## 12 2023-01-13 18:03:39.480 Eastern Pkwy & Washington Ave          3928.08
## 13 2023-01-10 10:07:00.410            E 14 St & Avenue B          5736.09
## 14 2023-01-08 11:51:15.057             E 2 St & Avenue A           5553.1
## 15 2023-01-08 03:14:36.628         7 Ave S & Bleecker St          5805.07
## 16 2023-01-02 15:59:17.992      Murray St & Greenwich St          5288.12
## 17 2023-01-08 19:02:52.270      Murray St & Greenwich St          5288.12
## 18 2023-01-03 19:11:30.449               E 47 St & 1 Ave          6498.09
## 19 2023-01-11 15:53:19.594       E 167 St & Franklin Ave          8048.01
## 20 2023-01-05 22:12:40.207             E 2 St & Avenue C          5476.03
##                end_station_name end_station_id start_lat start_lng  end_lat
## 1          Spruce St & Nassau St        5137.10  40.72486 -73.99213 40.71146
## 2                Ave A & E 11 St        5703.13  40.72486 -73.99213 40.72855
## 3                E 14 St & 1 Ave        5779.10  40.74714 -73.97113 40.73139
## 4      E Burnside Ave & Ryer Ave        8397.02  40.85054 -73.90132 40.85054
## 5        Clermont Ave & Park Ave        4692.01  40.69573 -73.97130 40.69573
## 6                Ave A & E 11 St        5703.13  40.73139 -73.98287 40.72855
## 7                Ave A & E 11 St        5703.13  40.74422 -73.97121 40.72855
## 8        Clermont Ave & Park Ave        4692.01  40.69985 -73.97141 40.69573
## 9        Clermont Ave & Park Ave        4692.01  40.69985 -73.97141 40.69573
## 10         Spruce St & Nassau St        5137.10  40.71780 -73.99316 40.71146
## 11    Underhill Ave & Lincoln Pl        4042.08  40.69985 -73.97141 40.67401
## 12             1 Pl & Clinton St        4193.14  40.67165 -73.96311 40.68096
## 13               E 14 St & 1 Ave        5779.10  40.72939 -73.97772 40.73139
## 14               Ave A & E 11 St        5703.13  40.72308 -73.98584 40.72855
## 15               Ave A & E 11 St        5703.13  40.73214 -74.00364 40.72855
## 16        W Broadway & Spring St        5569.06  40.71485 -74.01122 40.72495
## 17 North Moore St & Greenwich St        5470.12  40.71485 -74.01122 40.72020
## 18               E 14 St & 1 Ave        5779.10  40.75207 -73.96784 40.73139
## 19     Washington Ave & E 174 St        8277.03  40.82895 -73.90521 40.84308
## 20               Ave A & E 11 St        5703.13  40.72087 -73.98086 40.72855
##      end_lng member_casual
## 1  -74.00552        casual
## 2  -73.98176        casual
## 3  -73.98287        member
## 4  -73.90132        casual
## 5  -73.97130        casual
## 6  -73.98176        member
```

```
## 7  -73.98176        casual
## 8  -73.97130        casual
## 9  -73.97130        casual
## 10 -74.00552        member
## 11 -73.96715        member
## 12 -73.99906        member
## 13 -73.98287        member
## 14 -73.98176        member
## 15 -73.98176        member
## 16 -74.00166        casual
## 17 -74.01030        member
## 18 -73.98287        member
## 19 -73.90022        member
## 20 -73.98176        member
```

```
jan_1 <- jan_1 %>%
  drop_na() %>%
  select(
    -ride_id,
    -start_station_name, -start_station_id,
    -end_station_name, -end_station_id,
    -start_lat, -start_lng, -end_lat, -end_lng
  ) %>%
  mutate(
    started_at=ymd_hms(started_at),
    ended_at= ymd_hms(ended_at),

    start_date= as_date(started_at),
    end_date = as_date(ended_at),
    trip_duration_min = as.numeric(
      difftime(ended_at, started_at, units = "mins") #units in min!
    ),
    start_time = format(started_at, "%H:%M:%S"),
    end_time   = format(ended_at, "%H:%M:%S")
  ) %>%
  filter(start_date == end_date) %>% # drop overnight trips
  filter(trip_duration_min > 0,
         trip_duration_min < 1440) %>% # drop nonsensical durations
  mutate(
    start_date = format(start_date, "%m-%d") # drop year
  ) %>%
  select(
    -ended_at,
    -end_date,
    -started_at
  ) %>%
  arrange(start_date)

head(jan_1)
```

```
##   rideable_type member_casual start_date trip_duration_min start_time end_time
## 1 electric_bike        member      01-01          6.676250   12:36:20 12:43:01
## 2  classic_bike        member      01-01          8.699567   12:16:29 12:25:11
## 3  classic_bike        member      01-01         15.270617   15:11:37 15:26:53
```

```
## 4  classic_bike        casual      01-01         92.931400   16:37:10 18:10:06
## 5 electric_bike        member      01-01         11.120467   17:04:51 17:15:58
## 6  classic_bike        member      01-01          9.614283   19:37:49 19:47:25
```

```
jan_2 <- read.csv(file="~/Downloads/2023-citibike-tripdata/202301-citibike-tripdata_2.csv", header = T)

jan_2 <- jan_2 %>%
  drop_na() %>%
  select(
    -ride_id,
    -start_station_name, -start_station_id,
    -end_station_name, -end_station_id,
    -start_lat, -start_lng, -end_lat, -end_lng
  ) %>%
  mutate(
    started_at=ymd_hms(started_at),
    ended_at= ymd_hms(ended_at),

    start_date= as_date(started_at),
    end_date = as_date(ended_at),
    trip_duration_min = as.numeric(
      difftime(ended_at, started_at, units = "mins") #units in min!
    ),
    start_time = format(started_at, "%H:%M:%S"),
    end_time   = format(ended_at, "%H:%M:%S")
  ) %>%
  filter(start_date == end_date) %>%
  filter(trip_duration_min > 0,
         trip_duration_min < 1440) %>%
  mutate(
    start_date = format(start_date, "%m-%d") # drop year
  ) %>%
  select(
    -ended_at,
    -end_date,
    -started_at
  ) %>%
  arrange(start_date)

tail(jan_2)
```

```
##        rideable_type member_casual start_date trip_duration_min start_time
## 790495  classic_bike        member      01-31          2.117317   12:24:46
## 790496  classic_bike        member      01-31          2.678667   13:16:11
## 790497  classic_bike        casual      01-31          7.349983   22:27:46
## 790498 electric_bike        casual      01-31         13.541533   20:25:50
## 790499  classic_bike        member      01-31          4.856567   19:28:22
## 790500  classic_bike        member      01-31         10.306117   18:38:25
##        end_time
## 790495 12:26:53
## 790496 13:18:52
## 790497 22:35:07
## 790498 20:39:22
## 790499 19:33:13
```

```
## 790500 18:48:43
```

Binding the two dataset:

```
january <- bind_rows(jan_1, jan_2) %>%
  group_by(start_date) %>%
  summarise(
    trips = n(), # number of rides taken in a given day

    total_time = sum(trip_duration_min, na.rm = TRUE), #combined time traveled
    avg_time   = mean(trip_duration_min, na.rm = TRUE), # average time travel;ed

    member_percent = mean(member_casual == "member", na.rm = TRUE),
    casual_percent = mean(member_casual == "casual", na.rm = TRUE),

    ebike_percent   = mean(rideable_type == "electric_bike", na.rm = TRUE),
    regular_percent = mean(rideable_type == "classic_bike",  na.rm = TRUE),

    .groups = "drop"
  ) %>%
  mutate(
    member_percent  = 100 * member_percent,  # percentages of member
    casual_percent  = 100 * casual_percent,  # percentages of casual
    ebike_percent   = 100 * ebike_percent, # percentage of ebike
    regular_percent = 100 * regular_percent # percentage of regular
  ) %>%
  arrange(start_date)


head(january)
```

```
## # A tibble: 6 x 8
##   start_date trips total_time avg_time member_percent casual_percent
##   <chr>      <int>      <dbl>    <dbl>          <dbl>          <dbl>
## 1 01-01      50132    864227.     17.2           68.9           31.1
## 2 01-02      57912    838095.     14.5           76.6           23.4
## 3 01-03      51457    608820.     11.8           88.6           11.4
## 4 01-04      73955    932206.     12.6           85.8           14.2
## 5 01-05      71045    828095.     11.7           87.6           12.4
## 6 01-06      64353    738962.     11.5           86.6           13.4
## # i 2 more variables: ebike_percent <dbl>, regular_percent <dbl>
```

This must be repeated 11 times (for all the month of year), so a function:

```
files <- list.files(
  path = "~/Downloads/2023-citibike-tripdata/",
  pattern = "*.csv",
  full.names = TRUE
)
```

```
clean_file <- function(path) {
  read.csv(path, header = TRUE) %>%
```

```
    drop_na() %>%
    select(
      -ride_id,
      -start_station_name, -start_station_id,
      -end_station_name,   -end_station_id,
      -start_lat, -start_lng, -end_lat, -end_lng
    ) %>%
    mutate(
      started_at = ymd_hms(started_at),
      ended_at   = ymd_hms(ended_at),
      start_date = as_date(started_at),
      end_date   = as_date(ended_at),
      trip_duration_min = as.numeric(difftime(ended_at, started_at, units = "mins")),
      start_time = format(started_at, "%H:%M:%S"),
      end_time   = format(ended_at, "%H:%M:%S")
    ) %>%
    filter(start_date == end_date) %>%
    filter(trip_duration_min > 0, trip_duration_min < 1440) %>%
    mutate(
      start_date_full = start_date,
      # real date for sorting (and maybe if wwe later decide to do more year)
      start_date = format(start_date, "%m-%d")
    ) %>%
    select(-ended_at, -end_date, -started_at) %>%
    arrange(start_date_full) # no need of the whol year
}
```

```
files <- list.files(
  path = "~/Downloads/2023-citibike-tripdata/",
  pattern = "2023.*\\.csv$",
  full.names = TRUE,
  recursive = TRUE
)

all_2023 <- map_dfr(files, clean_file)
```

```
# making sure there is only two types for each
unique(all_2023$rideable_type)
```

```
## [1] "electric_bike" "classic_bike"
```

```
unique(all_2023$member_casual)
```

```
## [1] "member" "casual"
```

```
citibike_summary_2023 <- all_2023 %>%
  group_by(start_date_full) %>%
  summarise(
    trips = n(),
    total_time = sum(trip_duration_min, na.rm = TRUE),
    avg_time   = mean(trip_duration_min, na.rm = TRUE),
```

```
    member_percent = mean(member_casual == "member", na.rm = TRUE) * 100,
    casual_percent = mean(member_casual == "casual", na.rm = TRUE) * 100,

    ebike_percent   = mean(rideable_type == "electric_bike", na.rm = TRUE) * 100,
    regular_percent = mean(rideable_type == "classic_bike",  na.rm = TRUE) * 100,

    .groups = "drop"
  ) %>%
  arrange(start_date_full) %>%
  mutate(date = format(start_date_full, "%m-%d")) %>%
  select(date, everything(), -start_date_full)
tail(citibike_summary_2023)
```

```
## # A tibble: 6 x 8
##   date  trips total_time avg_time member_percent casual_percent ebike_percent
##   <chr> <int>      <dbl>    <dbl>          <dbl>          <dbl>         <dbl>
## 1 12-26 50418    639707.     12.7           81.6           18.4          63.9
## 2 12-27 42025    475598.     11.3           85.2           14.8          66.0
## 3 12-28 46324    562799.     12.1           84.1           15.9          65.8
## 4 12-29 68449    926314.     13.5           78.3           21.7          64.7
## 5 12-30 55573    703762.     12.7           78.2           21.8          65.5
## 6 12-31 53222    680897.     12.8           77.3           22.7          66.2
## # i 1 more variable: regular_percent <dbl>
```

```
#total number of trips
# total time (combined) in minutes
#average time (of the day) in minutes
# percent whether member or casual rider
# percent whether ebike or regular bike
```

```
write.csv(citibike_summary_2023,
          "~/Downloads/citibike_summary_2023.csv",
          row.names = FALSE)
```