Cyclistic Data Analysis - Data Processing

Gayatri Paul

2023-01-06

The data from October 2021 to October 2022 has been downloaded. Each file is in csv format. Let’s import it to R.

```
oct_21 <- read.csv("C:/Users/paulg/Downloads/csv/1_oct_21.csv", header=T, na.strings=c
("","NA"))

nov_21 <- read.csv("C:/Users/paulg/Downloads/csv/2_nov_21.csv", header=T, na.strings=c
("","NA"))

dec_21 <- read.csv("C:/Users/paulg/Downloads/csv/3_dec_21.csv", header=T, na.strings=c
("","NA"))

jan_22 <- read.csv("C:/Users/paulg/Downloads/csv/4_jan_22.csv", header=T, na.strings=c
("","NA"))

feb_22 = read.csv("C:/Users/paulg/Downloads/csv/5_feb_22.csv", header=T, na.strings=c("","
NA"))

mar_22 = read.csv("C:/Users/paulg/Downloads/csv/6_mar_22.csv", header=T, na.strings=c("","
NA"))

apr_22 = read.csv("C:/Users/paulg/Downloads/csv/7_apr_22.csv", header=T, na.strings=c("","
NA"))

may_22 = read.csv("C:/Users/paulg/Downloads/csv/8_may_22.csv", header=T, na.strings=c("","
NA"))

jun_22 = read.csv("C:/Users/paulg/Downloads/csv/9_jun_22.csv", header=T, na.strings=c("","
NA"))

jul_22 = read.csv("C:/Users/paulg/Downloads/csv/10_july_22.csv", header=T, na.strings=c
("","NA"))

aug_22 = read.csv("C:/Users/paulg/Downloads/csv/11_aug_22.csv", header=T, na.strings=c
("","NA"))

sept_22 = read.csv("C:/Users/paulg/Downloads/csv/12_sept_22.csv", header=T, na.strings=c
("","NA"))

oct_22 = read.csv("C:/Users/paulg/Downloads/csv/13_oct_22.csv", header=T, na.strings=c
("","NA"))
```

All the data frames contain 13 variables, lets

combine them for the purpose of analysis.

```
df= rbind(oct_21, nov_21, dec_21, jan_22, feb_22, mar_22, apr_22, may_22, jun_22, jul_22,
aug_22, sept_22, oct_22)
head(df)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 620BC6107255BF4C electric_bike 2021-10-22 12:46:42 2021-10-22 12:49:50
## 2 4471C70731AB2E45 electric_bike 2021-10-21 09:12:37 2021-10-21 09:14:14
## 3 26CA69D43D15EE14 electric_bike 2021-10-16 16:28:39 2021-10-16 16:36:26
## 4 362947F0437E1514 electric_bike 2021-10-16 16:17:48 2021-10-16 16:19:03
## 5 BB731DE2F2EC51C5 electric_bike 2021-10-20 23:17:54 2021-10-20 23:26:10
## 6 7176307BBC097313 electric_bike 2021-10-21 16:57:37 2021-10-21 17:11:58
##         start_station_name start_station_id end_station_name end_station_id
## 1 Kingsbury St & Kinzie St     KA1503000043             <NA>           <NA>
## 2                     <NA>             <NA>             <NA>           <NA>
## 3                     <NA>             <NA>             <NA>           <NA>
## 4                     <NA>             <NA>             <NA>           <NA>
## 5                     <NA>             <NA>             <NA>           <NA>
## 6                     <NA>             <NA>             <NA>           <NA>
##   start_lat start_lng end_lat end_lng member_casual
## 1  41.88919  -87.6385   41.89  -87.63        member
## 2  41.93000  -87.7000   41.93  -87.71        member
## 3  41.92000  -87.7000   41.94  -87.72        member
## 4  41.92000  -87.6900   41.92  -87.69        member
## 5  41.89000  -87.7100   41.89  -87.69        member
## 6  41.89000  -87.7100   41.93  -87.70        member
```

```
print(nrow(df))
```

```
## [1] 6386920
```

```
colnames(df)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"        
##  [4] "ended_at"           "start_station_name" "start_station_id"  
##  [7] "end_station_name"   "end_station_id"     "start_lat"         
## [10] "start_lng"          "end_lat"            "end_lng"           
## [13] "member_casual"
```

```
summary(df)
```

```
##    ride_id          rideable_type       started_at          ended_at        
##  Length:6386920     Length:6386920     Length:6386920     Length:6386920    
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##  start_station_name start_station_id   end_station_name   end_station_id    
##  Length:6386920     Length:6386920     Length:6386920     Length:6386920    
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##    start_lat       start_lng         end_lat         end_lng      
##  Min.   :41.64   Min.   :-87.84   Min.   :41.39   Min.   :-88.97  
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66  
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64  
##  Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65  
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63  
##  Max.   :45.64   Max.   :-73.80   Max.   :42.37   Max.   :-87.30  
##                                   NA's   :6319    NA's   :6319    
##  member_casual     
##  Length:6386920    
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
##                    
```

Rename the column named “member_casual” to “rider_type” for clarity.

```
library(dplyr)
df=rename(df,"rider_type"="member_casual")
head(df)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 620BC6107255BF4C electric_bike 2021-10-22 12:46:42 2021-10-22 12:49:50
## 2 4471C70731AB2E45 electric_bike 2021-10-21 09:12:37 2021-10-21 09:14:14
## 3 26CA69D43D15EE14 electric_bike 2021-10-16 16:28:39 2021-10-16 16:36:26
## 4 362947F0437E1514 electric_bike 2021-10-16 16:17:48 2021-10-16 16:19:03
## 5 BB731DE2F2EC51C5 electric_bike 2021-10-20 23:17:54 2021-10-20 23:26:10
## 6 7176307BBC097313 electric_bike 2021-10-21 16:57:37 2021-10-21 17:11:58
##         start_station_name start_station_id end_station_name end_station_id
## 1 Kingsbury St & Kinzie St     KA1503000043             <NA>           <NA>
## 2                     <NA>             <NA>             <NA>           <NA>
## 3                     <NA>             <NA>             <NA>           <NA>
## 4                     <NA>             <NA>             <NA>           <NA>
## 5                     <NA>             <NA>             <NA>           <NA>
## 6                     <NA>             <NA>             <NA>           <NA>
##   start_lat start_lng end_lat end_lng rider_type
## 1  41.88919  -87.6385   41.89  -87.63     member
## 2  41.93000  -87.7000   41.93  -87.71     member
## 3  41.92000  -87.7000   41.94  -87.72     member
## 4  41.92000  -87.6900   41.92  -87.69     member
## 5  41.89000  -87.7100   41.89  -87.69     member
## 6  41.89000  -87.7100   41.93  -87.70     member
```

Remove duplicate Rows

```
df=df[!duplicated(df), ]
print(nrow(df))
```

```
## [1] 6386920
```

Number of rows remained the same, which indicates that the data frame contained no duplicate data.

Find columns with missing information.

```
colSums(is.na(df))
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##             986387             986387            1054844            1054844
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0               6319               6319
##         rider_type
##                  0
```

Columns namely: start_station_name, start_station_id, end_station_name, end_station_id, end_lat, end_lng contain missing values. As we are not using these columns for analysis, we will keep them as they are. If we decide to use these columns, we will first remove rows with NA and then proceed further.

Convert date columns to proper date formats &

split date and time into 2 columns.

```r
library(lubridate)

df$started_at <- ymd_hms(df$started_at)
df$ended_at <- ymd_hms(df$ended_at)

df$start_date <- as.Date(df$started_at)
df$start_time <- format(as.POSIXct(df$started_at), format = "%H:%M:%S")

df$end_date <- as.Date(df$ended_at)
df$end_time <- format(as.POSIXct(df$ended_at), format = "%H:%M:%S")
```

Calculate ride length in mins

```r
df$ride_length <- round(difftime(df$ended_at,df$started_at, units = "mins"), 2)

df$ride_length <- as.numeric(df$ride_length)
```

keep rows with only positive ride lengths

```r
df <- filter(df, ride_length > 0)
head(df)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 620BC6107255BF4C electric_bike 2021-10-22 12:46:42 2021-10-22 12:49:50
## 2 4471C70731AB2E45 electric_bike 2021-10-21 09:12:37 2021-10-21 09:14:14
## 3 26CA69D43D15EE14 electric_bike 2021-10-16 16:28:39 2021-10-16 16:36:26
## 4 362947F0437E1514 electric_bike 2021-10-16 16:17:48 2021-10-16 16:19:03
## 5 BB731DE2F2EC51C5 electric_bike 2021-10-20 23:17:54 2021-10-20 23:26:10
## 6 7176307BBC097313 electric_bike 2021-10-21 16:57:37 2021-10-21 17:11:58
##         start_station_name start_station_id end_station_name end_station_id
## 1 Kingsbury St & Kinzie St     KA1503000043             <NA>           <NA>
## 2                     <NA>             <NA>             <NA>           <NA>
## 3                     <NA>             <NA>             <NA>           <NA>
## 4                     <NA>             <NA>             <NA>           <NA>
## 5                     <NA>             <NA>             <NA>           <NA>
## 6                     <NA>             <NA>             <NA>           <NA>
##   start_lat start_lng end_lat end_lng rider_type start_date start_time
## 1  41.88919  -87.6385   41.89  -87.63     member 2021-10-22   12:46:42
## 2  41.93000  -87.7000   41.93  -87.71     member 2021-10-21   09:12:37
## 3  41.92000  -87.7000   41.94  -87.72     member 2021-10-16   16:28:39
## 4  41.92000  -87.6900   41.92  -87.69     member 2021-10-16   16:17:48
## 5  41.89000  -87.7100   41.89  -87.69     member 2021-10-20   23:17:54
## 6  41.89000  -87.7100   41.93  -87.70     member 2021-10-21   16:57:37
##     end_date end_time ride_length
## 1 2021-10-22 12:49:50        3.13
## 2 2021-10-21 09:14:14        1.62
## 3 2021-10-16 16:36:26        7.78
## 4 2021-10-16 16:19:03        1.25
## 5 2021-10-20 23:26:10        8.27
## 6 2021-10-21 17:11:58       14.35
```

There are rides which lasted less than a minute which seems odd but, but as we don't have relevant data or source of information which can be used to confirm that these entries are wrong, we will assume that these are correct and continue with the analysis.

Calculate number of days the ride lasted

```
df$no_of_days <- as.numeric(difftime(df$end_date, df$start_date, units = "days" )+1)

count(filter(df, no_of_days > 1))
```

```
##       n
## 1 38417
```

38417 rides lasted more than a day. If we assume this data is correct, these users had to be charged extra, which means their ride must costed a lot, this case can be used to advocate annual memberships.

Calculate day of week for the ride

```
df$day_of_week_num <- wday(df$start_date)

df <- df %>%
  mutate(day_of_week=
           ifelse(day_of_week_num==1,"sunday",
                  ifelse(day_of_week_num==2, "monday",
                         ifelse(day_of_week_num==3, "tuesday",
                                ifelse(day_of_week_num==4, "wednesday",
                                       ifelse(day_of_week_num==5, "thursday",
                                              ifelse(day_of_week_num==6, "friday",
                                                     "saturday" )))))))
```

Add month and Season columns.

```
library(tidyverse)

df$month_col <- month(df$started_at)

df <- df %>%
  mutate(season=
           ifelse(month_col==12 |month_col==1 |month_col==2,
                  "winter",
                  ifelse(month_col==3 |month_col==4 |month_col==5,
                         "spring",
                         ifelse(month_col==6 |month_col==7 |month_col==8,
                                "summer",
                                "fall"))))

df$year <- year(df$started_at)

df$month_year= paste(df$month_col, df$year)

head(df)
```

```
##             ride_id rideable_type          started_at            ended_at
## 1 620BC6107255BF4C electric_bike 2021-10-22 12:46:42 2021-10-22 12:49:50
## 2 4471C70731AB2E45 electric_bike 2021-10-21 09:12:37 2021-10-21 09:14:14
## 3 26CA69D43D15EE14 electric_bike 2021-10-16 16:28:39 2021-10-16 16:36:26
## 4 362947F0437E1514 electric_bike 2021-10-16 16:17:48 2021-10-16 16:19:03
## 5 BB731DE2F2EC51C5 electric_bike 2021-10-20 23:17:54 2021-10-20 23:26:10
## 6 7176307BBC097313 electric_bike 2021-10-21 16:57:37 2021-10-21 17:11:58
##          start_station_name start_station_id end_station_name end_station_id
## 1 Kingsbury St & Kinzie St     KA1503000043             <NA>           <NA>
## 2                     <NA>             <NA>             <NA>           <NA>
## 3                     <NA>             <NA>             <NA>           <NA>
## 4                     <NA>             <NA>             <NA>           <NA>
## 5                     <NA>             <NA>             <NA>           <NA>
## 6                     <NA>             <NA>             <NA>           <NA>
##   start_lat start_lng end_lat end_lng rider_type start_date start_time
## 1  41.88919  -87.6385   41.89  -87.63     member 2021-10-22   12:46:42
## 2  41.93000  -87.7000   41.93  -87.71     member 2021-10-21   09:12:37
## 3  41.92000  -87.7000   41.94  -87.72     member 2021-10-16   16:28:39
## 4  41.92000  -87.6900   41.92  -87.69     member 2021-10-16   16:17:48
## 5  41.89000  -87.7100   41.89  -87.69     member 2021-10-20   23:17:54
## 6  41.89000  -87.7100   41.93  -87.70     member 2021-10-21   16:57:37
##     end_date end_time ride_length no_of_days day_of_week_num day_of_week
## 1 2021-10-22 12:49:50        3.13          1               6      friday
## 2 2021-10-21 09:14:14        1.62          1               5    thursday
## 3 2021-10-16 16:36:26        7.78          1               7    saturday
## 4 2021-10-16 16:19:03        1.25          1               7    saturday
## 5 2021-10-20 23:26:10        8.27          1               4   wednesday
## 6 2021-10-21 17:11:58       14.35          1               5    thursday
##   month_col season year month_year
## 1        10   fall 2021    10 2021
## 2        10   fall 2021    10 2021
## 3        10   fall 2021    10 2021
## 4        10   fall 2021    10 2021
## 5        10   fall 2021    10 2021
## 6        10   fall 2021    10 2021
```

Add holiday column.

```
df <- df %>%
  mutate(holiday=
           ifelse(start_date=="2021-11-11" |start_date=="2021-11-25"|start_date=="2021-12-
24"
                  |start_date=="2021-12-31"|start_date=="2022-01-01"|start_date=="2022-03-
17"
                  |start_date=="2022-05-30"|start_date=="2022-07-04"|start_date=="2022-09-
05"
                  |day_of_week=="sunday"|day_of_week=="saturday",
                  "holiday",
                  "workday"))
```

Next step would be to start analysing.