Bike Trips analysis

Musa Karimli

3/7/2022

Data importing

Because data was too large for my memory to handle, I exported CSV files to a local SQL server(Postgres), then imported data from here.

Let's first import data from sql:

```r
library(tidyverse)
library('RPostgreSQL')
library(dplyr)

con <- dbConnect(drv =PostgreSQL(),
                 user="postgres",
                 password="sinif555a",
                 host="localhost",
                 port=5432,
                 dbname="customer_data")

dbListTables(con)   #list all the tables
```

```
##  [1] "customer_purchase" "workbook"          "movies"
##  [4] "customer"          "nation"            "orders"
##  [7] "part"              "region"            "customer_address"
## [10] "employees"         "departments"       "automobile_data"
## [13] "warehouse"         "orders"            "avocado"
## [16] "tripdata"
```

```r
# query the order table
trips <- dbGetQuery(con, "SELECT * from trip.tripdata;")

trips<-as_tibble(trips)

glimpse(trips)
```

```
## Rows: 5,717,608
## Columns: 13
## $ ride_id            <chr> "89E7AA6C29227EFF", "0FEFDE2603568365", "E6159D746B~
## $ rideable_type      <chr> "classic_bike", "classic_bike", "electric_bike", "c~
## $ started_at         <chr> "2021-02-12 16:14:56", "2021-02-14 17:52:38", "2021~
## $ ended_at           <chr> "2021-02-12 16:21:43", "2021-02-14 18:12:09", "2021~
## $ start_station_name <chr> "Glenwood Ave & Touhy Ave", "Glenwood Ave & Touhy A~
## $ start_station_id   <chr> "525", "525", "KA1503000012", "637", "13216", "1800~
## $ end_station_name   <chr> "Sheridan Rd & Columbia Ave", "Bosworth Ave & Howar~
## $ end_station_id     <chr> "660", "16806", "TA1305000029", "TA1305000034", "TA~
## $ start_lat          <dbl> 42.01270, 42.01270, 41.88579, 41.89563, 41.83473, 4~
## $ start_lng          <dbl> -87.66606, -87.66606, -87.63110, -87.67207, -87.625~
## $ end_lat            <dbl> 42.00458, 42.01954, 41.88487, 41.90312, 41.83816, 4~
## $ end_lng            <dbl> -87.66141, -87.66956, -87.62750, -87.67394, -87.645~
## $ member_casual      <chr> "member", "casual", "member", "member", "member", "~
```

```r
summary(trips)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:5717608     Length:5717608     Length:5717608     Length:5717608
##  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:5717608     Length:5717608     Length:5717608     Length:5717608
##  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.   :42.17   Max.   :-87.49
##                                   NA's   :4831    NA's   :4831
##  member_casual
##  Length:5717608
##  Class :character
##  Mode  :character
##
##
##
##
```

Data cleaning and transformation

1. Dealing With date and time:

```r
library(lubridate)

# converting string to datetime
trips$ended_at <- ymd_hms(trips$ended_at)
trips$started_at <- ymd_hms(trips$started_at)

# calculating trip duration
trips['trip_duration']=trips$ended_at-trips$started_at

# excluding trips which are lasted zero seconds or below
trips <- trips %>% filter(trip_duration>0)

# extracting date components
trips$start_year <- year(trips$started_at)
trips$start_month <- month(trips$started_at)
trips$start_quarter <- quarter(trips$started_at)
trips$start_week <- week(trips$started_at)
trips$start_wday <- wday(trips$started_at)
trips$start_day <- day(trips$started_at)
trips$start_hour <- hour(trips$started_at)
trips$start_year_month <- floor_date(as_date(trips$started_at),"month")
```

2. Dealing with null and duplicate values:

```r
# replacing NA stations names with no info
trips <- trips %>%
    mutate(start_station_name = case_when(start_station_name==''~'No info',
                                          is.na(start_station_name)~'No info',
                                          TRUE ~start_station_name))%>%
             mutate(start_station_name = case_when(end_station_name==''~'No info',
                                        is.na(end_station_name)~'No info',
                                        TRUE ~end_station_name))


# there isn't any duplicate in "ride_id" column
trips %>% count(ride_id) %>%
    filter(n>1)
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: ride_id <chr>, n <int>
```

3. Cleaning String Data:

```r
library(stringr)

trips <- trips %>%
    mutate(start_station_name = str_trim(start_station_name,side='both')) %>%
    mutate(end_station_name = str_trim(end_station_name,side='both')) %>%
    mutate(start_station_name = str_to_title(start_station_name)) %>%
    mutate(end_station_name = str_to_title(end_station_name))
```

4. Adding additonal columns for data analysis:

```r
library("geosphere")
# finding the shortest distance between two locations
trips <- trips %>% mutate(distance_ctd = distHaversine(cbind(start_lng, start_lat),
                                                       cbind(end_lng, end_lat)))
# finding approximate speed of the ride
trips<-transform(trips,speed=distance_ctd/as.double(trip_duration,units='secs'))

# setting abnormal values to NA so it won't affect to the calculations
trips <- trips %>%
    mutate(speed = case_when(speed==0 ~ NA_real_,
                             is.infinite(speed) ~ NA_real_,
                             TRUE ~ as.numeric(speed)))

trips <- trips %>%
    mutate(distance_ctd = case_when(distance_ctd==0 ~ NA_real_,
                              is.infinite(distance_ctd) ~ NA_real_,
                              TRUE ~ as.numeric(distance_ctd)))
```

Data analysis

Looking at the data with summarization:

```r
trips %>% group_by(member_casual,rideable_type) %>%
    summarise("avg_duration_trip"=mean(trip_duration,na.rm=T),
              "avg_trip_distance"=mean(distance_ctd,na.rm=TRUE),
              "avg_speed"=mean(speed,na.rm=T),
              "n_of_rides"=format(n(),scientific=F))
```

```
## # A tibble: 5 x 6
## # Groups:   member_casual [2]
##   member_casual rideable_type avg_duration_trip avg_trip_distance avg_speed
##   <chr>         <chr>         <drtn>                        <dbl>     <dbl>
## 1 casual        classic_bike  1738.8018 secs                2360.      2.30
## 2 casual        docked_bike   4929.1841 secs                2548.      1.64
## 3 casual        electric_bike 1183.2579 secs                2627.      3.10
## 4 member        classic_bike   845.7443 secs                2084.      2.87
## 5 member        electric_bike  753.2742 secs                2461.      3.70
## # ... with 1 more variable: n_of_rides <chr>
```

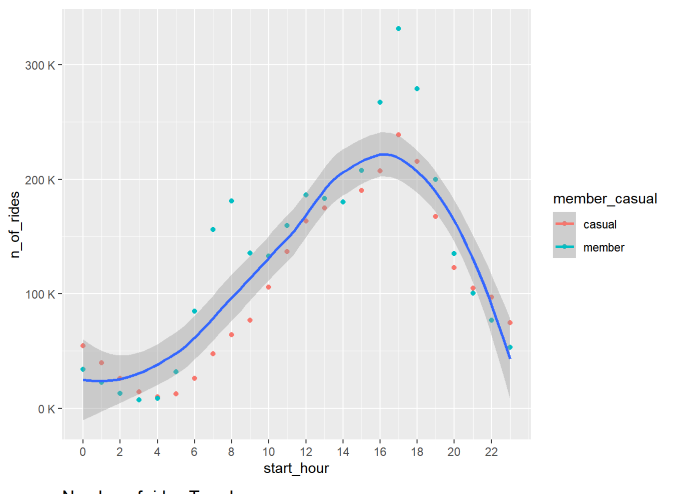
```r
trips %>% group_by(start_year,start_quarter,start_month) %>%
    summarise("avg_duration_trip"=mean(trip_duration,na.rm=T),
              "avg_trip_distance"=mean(distance_ctd,na.rm=TRUE),
              "avg_speed"=mean(speed,na.rm=T),
              "n_of_rides"=format(n(),scientific=F))
```

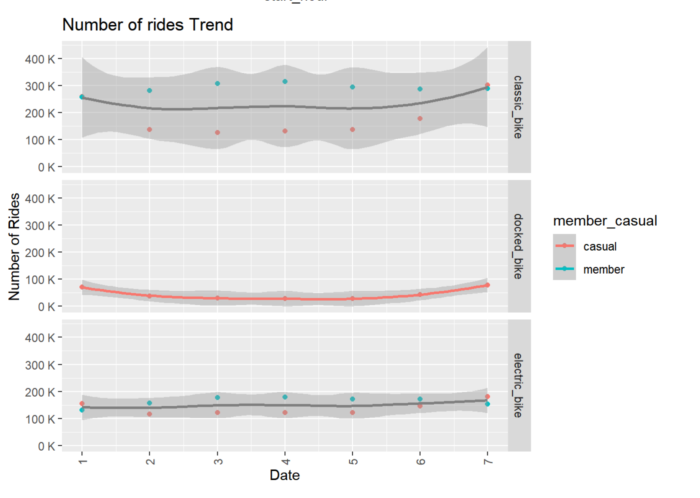
```
## # A tibble: 13 x 7
## # Groups:   start_year, start_quarter [5]
##    start_year start_quarter start_month avg_duration_trip avg_trip_distance
##         <dbl>         <int>       <dbl> <drtn>                        <dbl>
##  1       2021             1           2 1465.4887 secs                2134.
##  2       2021             1           3 1372.0176 secs                2299.
##  3       2021             2           4 1448.6085 secs                2332.
##  4       2021             2           5 1562.5566 secs                2381.
##  5       2021             2           6 1564.9780 secs                2431.
##  6       2021             3           7 1452.7834 secs                2447.
##  7       2021             3           8 1298.2863 secs                2437.
##  8       2021             3           9 1230.8146 secs                2420.
##  9       2021             4          10 1145.6809 secs                2260.
## 10       2021             4          11  889.3272 secs                2062.
## 11       2021             4          12  871.4834 secs                2039.
## 12       2022             1           1  915.9240 secs                1877.
## 13       2022             1           2  854.4562 secs                1931.
## # ... with 2 more variables: avg_speed <dbl>, n_of_rides <chr>
```

```r
trips %>% group_by(start_year,start_quarter,start_month,member_casual) %>%
    summarise("avg_duration_trip"=mean(trip_duration,na.rm=T),
              "avg_trip_distance"=mean(distance_ctd,na.rm=TRUE),
              "avg_speed"=mean(speed,na.rm=T),
              "n_of_rides"=format(n(),scientific=F))
```

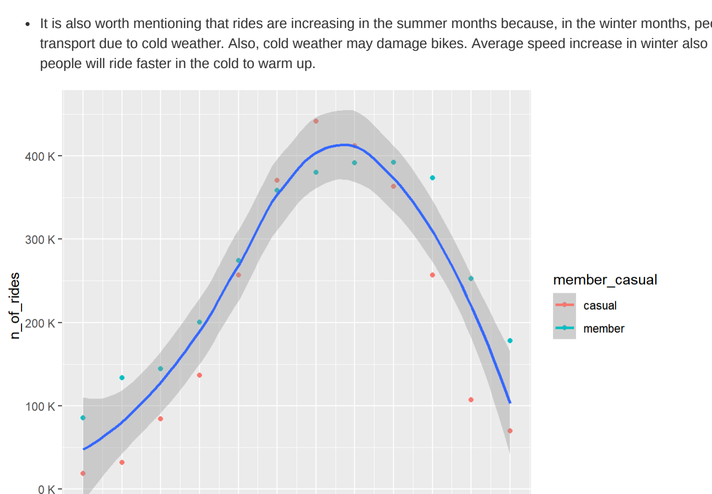
```
## # A tibble: 26 x 8
## # Groups:   start_year, start_quarter, start_month [13]
##    start_year start_quarter start_month member_casual avg_duration_trip
##         <dbl>         <int>       <dbl> <chr>         <drtn>
##  1       2021             1           2 casual        2962.6862 secs
##  2       2021             1           2 member        1081.4072 secs
##  3       2021             1           3 casual        2289.6601 secs
##  4       2021             1           3 member         838.2379 secs
##  5       2021             2           4 casual        2281.5631 secs
##  6       2021             2           4 member         881.4493 secs
##  7       2021             2           5 casual        2294.1080 secs
##  8       2021             2           5 member         878.4177 secs
##  9       2021             2           6 casual        2227.5568 secs
## 10       2021             2           6 member         880.7195 secs
## # ... with 16 more rows, and 3 more variables: avg_trip_distance <dbl>,
## #   avg_speed <dbl>, n_of_rides <chr>
```

- We can see that members ride bikes faster and shorter rides than casual riders.
- It is clear that members use a bike to commute to work daily. Also, exact start and end locations indicate that people didn't ride bikes to work—these values are not considered in several below analyses.
- Casual riders move more on weekends because of additional free time.
- The graph below supports this theory by showing that members' rides are much more than casuals' at the beginning and end of working hours.





- It is also worth mentioning that rides are increasing in the summer months because, in the winter months, people will likely use alternative transport due to cold weather. Also, cold weather may damage bikes. Average speed increase in winter also backs this theory because people will ride faster in the cold to warm up.



- Looking at members' rides' duration and distance at peak hours, I can find how many casual riders have a similar type of ride.

```r
# finding average duration and distance of members' rides at peak hours

trips %>% filter(((start_hour>=8)&(start_hour<=10))
                 |((start_hour>=16)&(start_hour<=18))
                 &(member_casual=='member')) %>%
                    summarise('average_dur' = mean(trip_duration,na.rm=T),
                              'average_dist' = mean(distance_ctd,na.rm=T))
```

```
##     average_dur average_dist
## 1 963.6844 secs     2287.052
```

```r
# finding similar type of casual rides
casual_rides <- trips %>%
    filter((start_hour>=8&start_hour<=10)
                 |(start_hour>=16&start_hour<=18)) %>%
               filter((member_casual=='casual')&(distance_ctd<=2287)
                 &(trip_duration<duration(964,units = 'second')))

cat('Casual rides in peak hours:', nrow(casual_rides),sep = ' ')
```

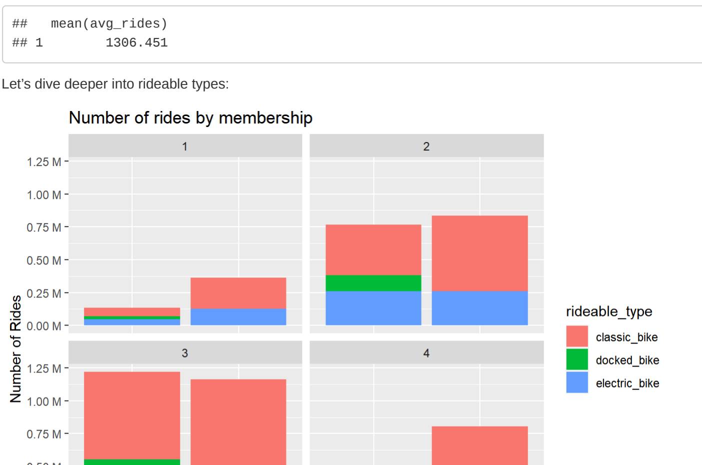
```
## Casual rides in peak hours: 344903
```

```r
# Assuming those casual riders used bikes for commuting to work, dividing the average of these rides by working d
ays, we could know how many casual riders we are missing from membership.

(casual_rides %>%
group_by(start_month,member_casual) %>%
    summarise("avg_duration_trip"=mean(trip_duration,na.rm=T),
              "avg_trip_distance"=mean(distance_ctd,na.rm=TRUE),
              "avg_speed"=mean(speed,na.rm=T),
              "n_of_rides"=as.numeric(format(n(),scientific=F)))%>%
    summarise('avg_rides'=mean(n_of_rides)) %>%
    summarise(mean(avg_rides)))/22
```

```
##   mean(avg_rides)
## 1        1306.451
```

Let's dive deeper into rideable types:



- It is evident that people prefer classic bikes. Docked bikes seem to be a new type of ride, electric bikes are less popular than classic bikes, but it has the potential to become more popular, we can see in the trend line.

Wrap Up

I would propose incentivizing individuals to use cycles as daily transport to commute. Acknowledging that using bikes reduces carbon emissions and traffic jams will help people use them. There are still people who don't subscribe as members who use bicycles. Campaigns, emails, and billboards may assist people in understanding the importance of lowering dependency on non-renewable energy.

Further, seasonal subscriptions, for example, making summer and winter plans differently, will draw more people in the summer to subscribe. But to expand membership in the winter, investing extra funds to have winter equipment to rent may benefit. Also, bikes need to be serviced accordingly. For example, bicycles may need better quality tires.

Investing in electric bikes may allow more people to commute because some people are also afraid of sweating, and it is less tiring than traditional bikes.