Case Study 1: Bike Trip Data

Rani Bhangu

2023-03-18

Installing the package Tidyverse

We require the package Tidyverse for any sort of Data operations in R. Installing and loading the library using the following:

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.0     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.1     v tibble    3.1.8
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become er
```

Reading the Data

Uploaded the Data file(CSV file) in R environment and then stored it into a dataframe called df by using the read_csv function

```
df=read_csv("202204divvytripdata.csv")
```

```
## Rows: 371249 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Summary of the Data

Initially we observed all the details about the columns. The total number of rows also was obtained from the summary.

```
summary(df)
```

```
##    ride_id          rideable_type        started_at                    
##  Length:371249      Length:371249      Min.   :2022-04-01 00:01:48.00  
##  Class :character   Class :character   1st Qu.:2022-04-10 12:44:18.00  
##  Mode  :character   Mode  :character   Median :2022-04-17 18:58:38.00  
##                                        Mean   :2022-04-17 05:02:35.16  
##                                        3rd Qu.:2022-04-23 20:44:14.00  
##                                        Max.   :2022-04-30 23:59:54.00  
##                                                                        
##     ended_at                       start_station_name start_station_id  
##  Min.   :2022-04-01 00:02:15.00   Length:371249      Length:371249     
##  1st Qu.:2022-04-10 13:05:50.00   Class :character   Class :character  
##  Median :2022-04-17 19:18:53.00   Mode  :character   Mode  :character  
##  Mean   :2022-04-17 05:20:13.27                                        
##  3rd Qu.:2022-04-23 21:12:25.00                                        
##  Max.   :2022-05-02 00:35:01.00                                        
##                                                                        
##  end_station_name   end_station_id        start_lat       start_lng     
##  Length:371249      Length:371249      Min.   :41.65   Min.   :-87.83  
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66  
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64  
##                                        Mean   :41.90   Mean   :-87.65  
##                                        3rd Qu.:41.93   3rd Qu.:-87.63  
##                                        Max.   :42.07   Max.   :-87.52  
##                                                                        
##     end_lat         end_lng       member_casual     
##  Min.   :41.63   Min.   :-87.85   Length:371249     
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character  
##  Median :41.90   Median :-87.64   Mode  :character  
##  Mean   :41.90   Mean   :-87.65                     
##  3rd Qu.:41.93   3rd Qu.:-87.63                     
##  Max.   :42.08   Max.   :-87.52                     
##  NA's   :317     NA's   :317
```

Segregate the member data

To seperate the annual members we create a new dataframe called df2 which will have only the member values from member_casual column and also we find the number of rows of df2

```
df2=df[df$member_casual=='member',]
nrow(df2)
```

```
## [1] 244832
```

Calculating the time taken for Bike rides

We create using the mutate function a new dataframe df3 containing the time difference column in hours. The time difference is taken by the difftime function using start time and end time of the rides

```
df3=mutate(df,time_difference_hours = difftime(ended_at, started_at, units = "hours"))
```

Analying the time differences obtained

We observed the time differences for all the rides and found the mean time taken for a ride and also the range i.e min and max values.

```
head(df3$time_difference_hours,50)
```

```
## Time differences in hours
##  [1] 0.196666667 0.336111111 0.102222222 0.156388889 0.094722222 0.071666667
##  [7] 0.077222222 0.206666667 0.008888889 0.024722222 0.229444444 0.058055556
## [13] 0.086666667 0.171388889 0.051666667 0.522500000 0.102500000 0.181944444
## [19] 0.106388889 0.090277778 0.088333333 0.920277778 0.192500000 0.153888889
## [25] 0.159722222 0.475277778 1.152500000 0.320277778 0.617777778 0.285277778
## [31] 0.185277778 0.940000000 0.072222222 0.439166667 0.085555556 0.241944444
## [37] 0.115555556 0.272777778 0.042222222 0.102222222 0.220833333 0.217500000
## [43] 0.183888889 0.048611111 0.113055556 0.099722222 0.077222222 0.076388889
## [49] 0.132500000 0.397500000
```

```
range(df3$time_difference_hours)
```

```
## Time differences in hours
## [1]   0.0000 352.0367
```

```
mean(df3$time_difference_hours)
```

```
## Time difference of 0.2939226 hours
```

Installing Geosphere

To calculate the distance between the start station and end station we use the disthaversine function which is a part of geosphere library. So installing the geosphere library.

```
install.packages("geosphere")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library(geosphere)
```

Creating a dataframe for the distances

Using disthaversine function we calculate the distance and we store it in the new dataframe called df4

```
df4=mutate(df,distance_km = distHaversine(cbind(start_lng, start_lat), cbind(end_lng, end_lat))*0.001)
```

Analyzing the ride distances

We observe the distance_km data as well as summarize it

```
head(df4$distance_km,50)
```

```
##  [1] 3.7630506035 3.4981000199 1.1330718748 1.6038612667 1.2479938241
##  [6] 0.7111707221 0.7111707221 1.5815192273 0.0005319243 0.0158869818
## [11] 1.1479289459 0.8429697757 0.7752734229 2.1072816827 0.3640807622
## [16] 6.4375231597 0.7337271336 2.0655161493 1.4325444455 0.7788775473
## [21] 0.7788775473 1.8966919646 2.0104534151 1.6877434224 1.7368665129
## [26] 6.3017996369 2.8997783052 6.1232253805 3.6674616349 4.9873860723
## [31] 2.4133772782 0.0000000000 0.0060484061 5.8765496517 1.2372846849
## [36] 3.6387956989 1.0616376423 1.5237146402 0.5339632434 1.3707625801
## [41] 1.7288941299 1.9865905289 3.3063146941 0.5289634540 1.5604722157
## [46] 0.9159067758 1.1536084023 1.1527754215 2.2454666603 1.6902702554
```

```
summary(df4$distance_km)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.830   1.465   2.041   2.646  28.608     317
```

Segergetating the casual data

we separate the casual data in the member_casual column and create it in data frame called df6

```
df6=df[df$member_casual=='casual',]
```

Analysing time difference

Using the mutate function we creat the time_difference_column for df6 by taking the difference between ended_at and started_at. We also view the time difference column in the new data frame created.

```
df7 <- mutate(df6, time_difference_hours = difftime(ended_at, started_at, units = "hours"))
head(df7$time_difference_hours,50)
```

```
## Time differences in hours
##  [1] 0.15638889 0.22944444 0.08666667 0.17138889 0.92027778 0.15388889
##  [7] 1.15250000 0.61777778 0.94000000 0.07222222 0.27277778 0.39750000
## [13] 0.47166667 0.09416667 0.33250000 0.37500000 0.17000000 0.72083333
## [19] 0.18444444 0.09361111 0.41861111 0.21222222 0.39000000 0.26527778
## [25] 1.10861111 0.16472222 0.94583333 0.59583333 0.23666667 0.21611111
## [31] 0.16527778 0.32277778 0.54333333 0.13944444 1.00305556 0.15944444
## [37] 0.22250000 0.08611111 0.10583333 0.09555556 0.09472222 0.16916667
## [43] 0.23027778 0.09250000 0.47055556 0.14500000 0.07138889 0.19750000
## [49] 0.44027778 0.05638889
```

Statistics of time differnce

For the data frame df7 which contains casual members and time diffrences for each of those rides we observe the range and the mean

```
range(df7$time_difference_hours)
```

```
## Time differences in hours
## [1]   0.0000 352.0367
```

```
mean(df7$time_difference_hours)
```

```
## Time difference of 0.4922071 hours
```

Observing the threashold value

We now observe for how many rides in the casual data the time difference is greater than a threshold value. After a lot of trials we took a threshold value of 5 for understanding how much percentage of casual riders can be converted to annual members.

```
sum(df7$time_difference_hours>5)
```

```
## [1] 541
```