GoogleCapstoneTT_Part2_FromRawDataToClean

TT

31/01/2022

R Markdown

This is **second** part of my capstone project. Ref to previous parts. In the first part data have been collected and browsed. Therafter data was manipulated and processed. As as result data from various csv files has been made compatible. Thereafter data was merged into one large dataframe and exported to a big CSV file for further analysis.

Loading required packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

```
library(ggplot2)
library(dplyr)
```

Loading previously created database:

```
bike_data_2019_raw <- read.csv("E:/Tomasz/CapstoneGoogle/capstone_bike_2019_base.csv")
```

Searching for missing values in some colums

```
(colMeans(is.na(bike_data_2019_raw)))
```

```
##                                           X
##                                   0.0000000
##                                     ride_id
##                                   0.0000000
##                                  started_at
##                                   0.0000000
##                                    ended_at
##                                   0.0000000
##                                rideable_type
##                                   0.0000000
##                                 tripduration
##                                   0.2902467
##                              start_station_id
##                                   0.0000000
##                            start_station_name
##                                   0.0000000
##                               end_station_id
##                                   0.0000000
##                             end_station_name
##                                   0.0000000
##                                member_casual
##                                   0.0000000
##                                      gender
##                                   0.3881125
##                                   birthyear
##                                   0.3839600
## X01...Rental.Details.Duration.In.Seconds.Uncapped
##                                   0.7097533
##                                 Member.Gender
##                                   0.7583531
##         X05...Member.Details.Member.Birthday.Year
##                                   0.7571480
```

While most of the colums contains no missing values there are some colums with significant percentage of missing data. This would require some actions like for example: collecting more data, replacing missing value with other value (like mean, max, median, 0 etc). For the purpose of this capstone I will drop the columns and focus on analysing other columns.

Removing not required colums and creating new dataframe

```
bike_data <- bike_data_2019_raw %>%
select(-c(birthyear, gender, "X01...Rental.Details.Duration.In.Seconds.Uncapped", "Member.Gender", "X05...Member.
Details.Member.Birthday.Year"))
```

Brief database statistical summary

```
summary(bike_data)  #Statistical summary of data. Mainly for numerics
```

```
##        X              ride_id          started_at          ended_at
##  Min.   :      1   Min.   :21742443   Length:3818004     Length:3818004
##  1st Qu.: 954502   1st Qu.:22873787   Class :character   Class :character
##  Median :1909002   Median :23962320   Mode  :character   Mode  :character
##  Mean   :1909002   Mean   :23915629
##  3rd Qu.:2863503   3rd Qu.:24963703
##  Max.   :3818004   Max.   :25962904
##
##  rideable_type   tripduration      start_station_id start_station_name
##  Min.   :   1   Min.   :      61   Min.   :  1.0    Length:3818004
##  1st Qu.:1727   1st Qu.:     405   1st Qu.: 77.0    Class :character
##  Median :3451   Median :     696   Median :174.0    Mode  :character
##  Mean   :3380   Mean   :    1500   Mean   :201.7
##  3rd Qu.:5046   3rd Qu.:    1257   3rd Qu.:289.0
##  Max.   :6946   Max.   :10628400   Max.   :673.0
##                 NA's   :1108163
##  end_station_id  end_station_name   member_casual
##  Min.   :  1.0   Length:3818004     Length:3818004
##  1st Qu.: 77.0   Class :character   Class :character
##  Median :174.0   Mode  :character   Mode  :character
##  Mean   :202.6
##  3rd Qu.:291.0
##  Max.   :673.0
##
```

Retrieving some basic dataframe info

```
dim(bike_data)
```

```
## [1] 3818004      11
```

```
nrow(bike_data)
```

```
## [1] 3818004
```

```
ncol(bike_data)
```

```
## [1] 11
```

```
colMeans(is.na(bike_data))
```

```
##                 X           ride_id         started_at           ended_at
##         0.0000000         0.0000000         0.0000000         0.0000000
##      rideable_type       tripduration   start_station_id start_station_name
##         0.0000000         0.2902467         0.0000000         0.0000000
##     end_station_id   end_station_name      member_casual
##         0.0000000         0.0000000         0.0000000
```

Insight No 1:

dimention(381804 x 11), number of rows (381804), number of columns (381804). No more missing values except column “tripduration”. New column with the length of the ride will therefore be created.

Column names and browsing first 6 rows of dataframe

```
colnames(bike_data)
```

```
##  [1] "X"                 "ride_id"           "started_at"
##  [4] "ended_at"          "rideable_type"     "tripduration"
##  [7] "start_station_id"  "start_station_name" "end_station_id"
## [10] "end_station_name"  "member_casual"
```

```
head(bike_data)
```

```
##   X  ride_id          started_at            ended_at rideable_type tripduration
## 1 1 21742443 2019-01-01 00:04:37 2019-01-01 00:11:07          2167          390
## 2 2 21742444 2019-01-01 00:08:13 2019-01-01 00:15:34          4386          441
## 3 3 21742445 2019-01-01 00:13:23 2019-01-01 00:27:12          1524          829
## 4 4 21742446 2019-01-01 00:13:45 2019-01-01 00:43:28           252         1783
## 5 5 21742447 2019-01-01 00:14:52 2019-01-01 00:20:56          1170          364
## 6 6 21742448 2019-01-01 00:15:33 2019-01-01 00:19:09          2437          216
##   start_station_id                  start_station_name end_station_id
## 1              199              Wabash Ave & Grand Ave             84
## 2               44              State St & Randolph St            624
## 3               15                Racine Ave & 18th St            644
## 4              123      California Ave & Milwaukee Ave            176
## 5              173 Mies van der Rohe Way & Chicago Ave             35
## 6               98          LaSalle St & Washington St             49
##                 end_station_name member_casual
## 1      Milwaukee Ave & Grand Ave    Subscriber
## 2 Dearborn St & Van Buren St (*)    Subscriber
## 3  Western Ave & Fillmore St (*)    Subscriber
## 4              Clark St & Elm St    Subscriber
## 5        Streeter Dr & Grand Ave    Subscriber
## 6        Dearborn St & Monroe St    Subscriber
```

#Finding unique values in column member_casual

```
unique(bike_data$member_casual)
```

```
## [1] "Subscriber" "Customer"
```

```
table(bike_data["member_casual"])
```

```
##
##   Customer Subscriber
##     880637    2937367
```

```
table(bike_data$member_casual)
```

```
##
##   Customer Subscriber
##     880637    2937367
```

Insight No 2:

Two unique customers types: Subscriber 2937367 observations, Customer 880637 observations.

Creating new colums which will allow to analyse datframe after aggregation and when it comes to date, month, weekday etc. (#)yyyy-mm-dd as default)

```
bike_data$date <- as.Date(bike_data$started_at)
bike_data$month <- format(as.Date(bike_data$date), "%m")
bike_data$day <- format(as.Date(bike_data$date), "%d")
bike_data$year <- format(as.Date(bike_data$date), "%Y")
bike_data$day_of_week <- format(as.Date(bike_data$date), "%A")
```

Converting data format from “chr” to “time” and creating “ride_length” column (in minutes).

```
str(bike_data)
```

```
## 'data.frame':    3818004 obs. of  16 variables:
##  $ X                 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ ride_id           : num  21742443 21742444 21742445 21742446 21742447 ...
##  $ started_at        : chr  "2019-01-01 00:04:37" "2019-01-01 00:08:13" "2019-01-01 00:13:23" "2019-01-01 00:1
3:45" ...
##  $ ended_at          : chr  "2019-01-01 00:11:07" "2019-01-01 00:15:34" "2019-01-01 00:27:12" "2019-01-01 00:4
3:28" ...
##  $ rideable_type     : int  2167 4386 1524 252 1170 2437 2708 2796 6205 3939 ...
##  $ tripduration      : int  390 441 829 1783 364 216 177 100 1727 336 ...
##  $ start_station_id  : int  199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr  "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave & 18th St" "Californ
ia Ave & Milwaukee Ave" ...
##  $ end_station_id    : int  84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr  "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "Western Ave & Fillmo
re St (*)" "Clark St & Elm St" ...
##  $ member_casual     : chr  "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ date              : Date, format: "2019-01-01" "2019-01-01" ...
##  $ month             : chr  "01" "01" "01" "01" ...
##  $ day               : chr  "01" "01" "01" "01" ...
##  $ year              : chr  "2019" "2019" "2019" "2019" ...
##  $ day_of_week       : chr  "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...
```

```
bike_data$started_at <- as_datetime(bike_data$started_at)
bike_data$ended_at <- as_datetime(bike_data$ended_at)

str(bike_data)
```

```
## 'data.frame':    3818004 obs. of  16 variables:
##  $ X                 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ ride_id           : num  21742443 21742444 21742445 21742446 21742447 ...
##  $ started_at        : POSIXct, format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ ended_at          : POSIXct, format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ rideable_type     : int  2167 4386 1524 252 1170 2437 2708 2796 6205 3939 ...
##  $ tripduration      : int  390 441 829 1783 364 216 177 100 1727 336 ...
##  $ start_station_id  : int  199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr  "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave & 18th St" "Californ
ia Ave & Milwaukee Ave" ...
##  $ end_station_id    : int  84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr  "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "Western Ave & Fillmo
re St (*)" "Clark St & Elm St" ...
##  $ member_casual     : chr  "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ date              : Date, format: "2019-01-01" "2019-01-01" ...
##  $ month             : chr  "01" "01" "01" "01" ...
##  $ day               : chr  "01" "01" "01" "01" ...
##  $ year              : chr  "2019" "2019" "2019" "2019" ...
##  $ day_of_week       : chr  "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...
```

```
bike_data$ride_length <- difftime(bike_data$ended_at,bike_data$started_at)
```

Export of data to new cleaned CSV file for further analysis.

write.csv(bike_data, file ='E:/Tomasz/CapstoneGoogle/capstone_bike_data_cleaned.csv')

Comment made to avoid consecutive exporting of this large file. Uncomment if required and copy cody to a chunk below