03 Process Data

S A Nawash Akhtar

Data cleaning processes of Cyclistic bike-share case study

1. Data description

In this document, data cleaning steps have been explained for the Cyclistic's raw data for future analysis processes. In the AWS server where the historical trip data is located, data can be found up to 2013. But for the purpose of this case study data from January 2020 to October 2022 will be analyzed. The analysis is done in November 2022. For reference, "Cyclistic" is a fictional company which is based on "Divvy Bikes" of Chicago.

2. Load packages

```
library(tidyverse)
library(data.table)
```

3. Create directory

```
my_path <- "E:\\Nawash\\Works\\Datasets\\Google data analysis\\Track 1\\CS1_Bike_share\\Csv files"
```

4. Find data type mismatch for the year 2020

Loading data for the year 2020 failed due to data type mismatch. So the data structure of the year 2020 have to be checked to find the mismatch.

```
glimpse(fread("Divvy_Trips_2020_Q1.csv"))
glimpse(fread("202004-divvy-tripdata.csv"))
glimpse(fread("202005-divvy-tripdata.csv"))
glimpse(fread("202006-divvy-tripdata.csv"))
glimpse(fread("202007-divvy-tripdata.csv"))
glimpse(fread("202008-divvy-tripdata.csv"))
glimpse(fread("202009-divvy-tripdata.csv"))
glimpse(fread("202010-divvy-tripdata.csv"))
glimpse(fread("202011-divvy-tripdata.csv"))
glimpse(fread("202012-divvy-tripdata.csv"))
```

5. Data type correction of 2020 year data.

Data type mismatch was found by thoroughly checking the data for the year 2020. Two variables that didn't match with the other data. So these two variable will be changed to the corrected data type.

```
setwd(paste(my_path, "\\2020", sep = ""))
dir()
q1_2020 <- fread("Divvy_Trips_2020_Q1.csv") %>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))
apr_2020 <- fread("202004-divvy-tripdata.csv") %>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))
may_2020 <- fread("202005-divvy-tripdata.csv") %>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))
jun_2020 <- fread("202006-divvy-tripdata.csv") %>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))
jul_2020 <- fread("202007-divvy-tripdata.csv") %>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))
aug_2020 <- fread("202008-divvy-tripdata.csv") %>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))
sep_2020 <- fread("202009-divvy-tripdata.csv") %>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))
oct_2020 <- fread("202010-divvy-tripdata.csv") %>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))
nov_2020 <- fread("202011-divvy-tripdata.csv") %>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))
dec_2020 <- fread("202012-divvy-tripdata.csv")
```

6. Combine dataset

```
trips_2020 <- bind_rows(q1_2020, apr_2020, may_2020, jun_2020,
                        jul_2020, aug_2020, sep_2020, oct_2020,
                        nov_2020, dec_2020)
```

6.1 Combine 2020 year data after correction

```
setwd(paste(my_path, "\\2021", sep = ""))
dir()
trips_2021 <- dir(full.names = TRUE) %>%
  map_df(fread)
```

6.2 Load 2021 data

```
setwd(paste(my_path, "\\2022", sep = ""))
dir()
trips_2022 <- dir(full.names = TRUE) %>%
  map_df(fread)
```

6.3 Load 2022 data

```
all_trips <- bind_rows(trips_2020, trips_2021, trips_2022)
```

6.4 Combine 2020, 2021 and 2022 dataset

```
glimpse(all_trips)
```

6.5 Checking dataset

```
## Rows: 14,284,922
## Columns: 13
## $ ride_id            <chr> "EACB19130B0CDA4A", "8FED874C809DC021", "789F3C21E4~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <dttm> 2020-01-21 20:06:59, 2020-01-30 14:22:39, 2020-01-~
## $ ended_at           <dttm> 2020-01-21 20:14:30, 2020-01-30 14:26:22, 2020-01-~
## $ start_station_name <chr> "Western Ave & Leland Ave", "Clark St & Montrose Av~
## $ start_station_id   <chr> "239", "234", "296", "51", "66", "212", "96", "96",~
## $ end_station_name   <chr> "Clark St & Leland Ave", "Southport Ave & Irving Pa~
## $ end_station_id     <chr> "326", "318", "117", "24", "212", "96", "212", "212~
## $ start_lat          <dbl> 41.9665, 41.9616, 41.9401, 41.8846, 41.8856, 41.889~
## $ start_lng          <dbl> -87.6884, -87.6660, -87.6455, -87.6319, -87.6418, -~
## $ end_lat            <dbl> 41.9671, 41.9542, 41.9402, 41.8918, 41.8899, 41.884~
## $ end_lng            <dbl> -87.6674, -87.6644, -87.6530, -87.6206, -87.6343, -~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

7. Creating new varible for ride duration

A new variable named ride duration will be created. It is basically the difference between the start time and the end time of the trip. The value of the variable will show in minutes.

```
all_trips$ride_duration <- difftime(all_trips$ended_at, all_trips$started_at,
                                    units = "mins")
```

Ride duration values will be converted to numeric for future analysis.

```
all_trips$ride_duration <- as.numeric(as.character(all_trips$ride_duration))
```

Checking the ride duration.

```
summary(all_trips$ride_duration)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -29049.97      6.63     11.90     21.96     21.78 156450.40
```

Some negative ride duration values are found which are not possible. Negative values will be removed for future analysis.

```
all_trips_v2 <- all_trips %>%
  filter(!(ride_duration < 0 ))
```

Checking ride duration again after removing negative values.

```
summary(all_trips_v2$ride_duration)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      0.00      6.63     11.90     22.73     21.80 156450.40
```

8. Create new date and time related variables for future analysis

For each ride, the datetime variable "started_at" will be separated into multiple variables for future use.

```
all_trips_v2$day <- as.integer(format(all_trips_v2$started_at, "%d"))
all_trips_v2$month <- as.integer(format(all_trips_v2$started_at, "%m"))
all_trips_v2$year <- as.integer(format(all_trips_v2$started_at, "%Y"))
all_trips_v2$day_of_week <- format(all_trips_v2$started_at, "%A")
all_trips_v2$date_ymd <- format(all_trips_v2$started_at, "%Y-%m-%d")
all_trips_v2$time_of_ride <- format(all_trips_v2$started_at, "%H:%M:%S")
```

Sorting the dataset by date of ride.

```
all_trips_v2 <- all_trips_v2 %>%
  arrange(all_trips_v2$date_ymd)
```

9. Remove blank and duplicate values

```
table(all_trips_v2$start_station_name)
table(all_trips_v2$end_station_name)
```

9.1 Checking start sation and end station names

```
all_trips_v3 <- all_trips_v2 %>%
  filter(!(is.na(start_station_name) | start_station_name == "")) %>%
  filter(!(is.na(end_station_name) | end_station_name == ""))
```

9.2 Remove blank values

9.2 Remove duplicate values Ride ID is unique for each ride so duplicate ride ID will be checked and deleted accordingly.

```
all_trips_v3$ride_id[duplicated(all_trips_v3$ride_id)] # Checking duplicate
all_trips_v4 <- all_trips_v3 %>% # Removing duplicate
  distinct(ride_id, .keep_all = TRUE)
```

```
glimpse(all_trips_v4)
```

9.3 Check cleaned dataset

```
## Rows: 11,945,997
## Columns: 20
## $ ride_id            <chr> "067126DC525B79F8", "2DC6C21EAE43DEDD", "6FB63011DA~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <dttm> 2020-01-01 17:05:43, 2020-01-01 15:07:07, 2020-01-~
## $ ended_at           <dttm> 2020-01-01 17:13:04, 2020-01-01 15:26:29, 2020-01-~
## $ start_station_name <chr> "Broadway & Belmont Ave", "Ada St & Washington Blvd~
## $ start_station_id   <chr> "296", "346", "81", "232", "227", "129", "76", "123~
## $ end_station_name   <chr> "Clark St & Drummond Pl", "Daley Center Plaza", "Ad~
## $ end_station_id     <chr> "220", "81", "346", "227", "254", "71", "255", "158~
## $ start_lat          <dbl> 41.9401, 41.8828, 41.8842, 41.9493, 41.9482, 41.857~
## $ start_lng          <dbl> -87.6455, -87.6612, -87.6296, -87.6463, -87.6639, -~
## $ end_lat            <dbl> 41.9312, 41.8842, 41.8828, 41.9482, 41.9544, 41.885~
## $ end_lng            <dbl> -87.6443, -87.6296, -87.6612, -87.6639, -87.6480, -~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
## $ ride_duration      <dbl> 7.350000, 19.366667, 14.466667, 8.316667, 8.466667,~
## $ day                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ month              <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ year               <int> 2020, 2020, 2020, 2020, 2020, 2020, 2020, 2020, 202~
## $ day_of_week        <chr> "Wednesday", "Wednesday", "Wednesday", "Wednesday",~
## $ date_ymd           <date> 2020-01-01, 2020-01-01, 2020-01-01, 2020-01-01, 20~
## $ time_of_ride       <chr> "17:05:43", "15:07:07", "18:25:13", "09:15:13", "10~
```

10. Save the cleaned dataset

Now cleaned dataset will be saved as csv file for future analysis.

```
setwd(my_path)
dir()
fwrite(all_trips_v4, "all_trips_4.csv")
```
