Cleaning Cylistic Data 2022-12

2023-07-31

Import data

```
data_01 <- read.csv(file="dataset/202212-divvy-tripdata.csv")
```

Check data 01

Check the data type for each meta

```
str(data_01)
```

```
## 'data.frame':    181806 obs. of  13 variables:
##  $ ride_id           : chr  "65DBD2F447EC51C2" "0C201AA7EA0EA1AD" "E0B148CCB358A49D" "54C5775D2B7C918
##  $ rideable_type     : chr  "electric_bike" "classic_bike" "electric_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-12-05 10:47:18" "2022-12-18 06:42:33" "2022-12-13 08:47:45" "2022-1
##  $ ended_at          : chr  "2022-12-05 10:56:34" "2022-12-18 07:08:44" "2022-12-13 08:59:51" "2022-1
##  $ start_station_name: chr  "Clifton Ave & Armitage Ave" "Broadway & Belmont Ave" "Sangamon St & Lak
##  $ start_station_id  : chr  "TA1307000163" "13277" "TA1306000015" "KA1503000038" ...
##  $ end_station_name  : chr  "Sedgwick St & Webster Ave" "Sedgwick St & Webster Ave" "St. Clair St & I
##  $ end_station_id    : chr  "13191" "13191" "13016" "13134" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.8 41.9 ...
##  $ start_lng         : num  -87.7 -87.6 -87.7 -87.6 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.6 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "member" "casual" "member" "member" ...
```

```
summary(data_01)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:181806      Length:181806      Length:181806      Length:181806
##  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:181806      Length:181806      Length:181806      Length:181806
##  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.   :41.65   Min.   :-87.83   Min.   :41.64   Min.   :-87.84
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.90   Median :-87.65   Median :41.90   Median :-87.65
##  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.   :42.07   Max.   :-87.53   Max.   :42.07   Max.   :-87.52
##                                   NA's   :128     NA's   :128
##  member_casual
##  Length:181806
##  Class :character
##  Mode  :character
##
##
##
##
```

From meta check we know that data type of column "started_at" and "end_at" should be datetime

Check duplicate data 01

```
print(data_01[duplicated(data_01), ])
```

```
##  [1] ride_id            rideable_type      started_at         ended_at
##  [5] start_station_name start_station_id   end_station_name   end_station_id
##  [9] start_lat          start_lng          end_lat            end_lng
## [13] member_casual
## <0 rows> (or 0-length row.names)
```

Duplicate data checking result : no data duplicate in data_01

Remove duplicate data

Remove Duplicate data result : No data to remove

Check missing value data in character data type

```
count(data_01[data_01$ride_id=="", ])
```

```
##   n
## 1 0
```

```
count(data_01[data_01$rideable_type=="", ])
```

```
##   n
## 1 0
```

```
count(data_01[data_01$started_at=="", ])
```

```
##   n
## 1 0
```

```
count(data_01[data_01$ended_at=="", ])
```

```
##   n
## 1 0
```

```
count(data_01[data_01$start_station_name=="", ])
```

```
##       n
## 1 29283
```

```
count(data_01[data_01$start_station_id=="", ])
```

```
##       n
## 1 29283
```

```
count(data_01[data_01$end_station_name=="", ])
```

```
##       n
## 1 31158
```

```
count(data_01[data_01$end_station_id=="", ])
```

```
##       n
## 1 31158
```

```
count(data_01[data_01$member_casual=="", ])
```

```
##   n
## 1 0
```

Missing value checking result :

ride_id: [0] rideable_type: [0] started_at: [0] ended_at: [0] start_station_name: [29,283] start_station_id: [29,283] end_station_name: [31,158] end_station_id: [31,158] member_casual: [0]

Fill Missing value with NA

Missing value (empty data) in start_station_name, start_station_id, end_station_name, end_station_id will be filling with NA

```r
data_01 <- replace(data_01, data_01 == "", NA)
```

Fill missing value result : empty data was replace with NA

Check missing value data

```r
count(data_01[is.na(data_01$start_lat) | data_01$start_lat=="", ])
```

```
##   n
## 1 0
```

```r
count(data_01[is.na(data_01$start_lng) | data_01$start_lng=="", ])
```

```
##   n
## 1 0
```

```r
count(data_01[is.na(data_01$end_lat) | data_01$end_lat=="", ])
```

```
##     n
## 1 128
```

```r
count(data_01[is.na(data_01$end_lng) | data_01$end_lng=="", ])
```

```
##     n
## 1 128
```

Missing value checking result :

start latitude and langitude : [0] end latitude and langitude : [128]

Remove Missing value with NA

Missing value in end_lat, end_lng will be delete by remove the row

```r
# remove missing value data in this other data if there are also missing values
# data_01 <- data_01[!is.na(data_01$rideable_type), ]
# data_01 <- data_01[!is.na(data_01$started_at), ]
# data_01 <- data_01[!is.na(data_01$ended_at), ]
# data_01 <- data_01[!is.na(data_01$member_casual), ]

data_01 <- data_01[!is.na(data_01$end_lat), ]
data_01 <- data_01[!is.na(data_01$end_lng), ]

count(data_01[is.na(data_01$end_lat) | data_01$end_lat=="", ])
```

```
##   n
## 1 0
```

```r
count(data_01[is.na(data_01$end_lng) | data_01$end_lng=="", ])
```

```
##   n
## 1 0
```

Remove missing value result : Row with missing value data was removed

Check outliers in coordinate data

```r
print(cat("start_lat : mean max min : ",
    mean(data_01$start_lat),
    max(data_01$start_lat),
    min(data_01$start_lat)))
```

```
## start_lat : mean max min :  41.90082 42.07 41.6485NULL
```

```r
print(cat("start_lng : mean max min : ",
    mean(data_01$start_lng), max(data_01$start_lng), min(data_01$start_lng)))
```

```
## start_lng : mean max min :  -87.64929 -87.52823 -87.83NULL
```

```r
print(cat("end_lat : mean max min : ",
    mean(data_01$end_lat), max(data_01$end_lat), min(data_01$end_lat)))
```

```
## end_lat : mean max min :  41.90093 42.07 41.64NULL
```

```r
print(cat("end_lng : mean max min : ",
    mean(data_01$end_lng), max(data_01$end_lng), min(data_01$end_lng)))
```

```
## end_lng : mean max min :  -87.6494 -87.52 -87.84NULL
```

Outliers checking result : no outliers in coordinate data, max and min value for each data doesnt far from average value

Remove useless column data

Acording to the bussines task, start_station_name and end_station_name will be remove

```r
data_01 <- data_01[, -which(names(data_01) == "start_station_name")]
data_01 <- data_01[, -which(names(data_01) == "end_station_name")]

head(data_01)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 65DBD2F447EC51C2 electric_bike 2022-12-05 10:47:18 2022-12-05 10:56:34
## 2 0C201AA7EA0EA1AD  classic_bike 2022-12-18 06:42:33 2022-12-18 07:08:44
## 3 E0B148CCB358A49D electric_bike 2022-12-13 08:47:45 2022-12-13 08:59:51
## 4 54C5775D2B7C9188  classic_bike 2022-12-13 18:50:47 2022-12-13 19:19:48
## 5 A4891F78776D35DF  classic_bike 2022-12-14 16:13:39 2022-12-14 16:27:50
## 6 DB91D9B8DFACA07A electric_bike 2022-12-02 15:24:47 2022-12-02 15:34:14
##   start_station_id end_station_id start_lat start_lng  end_lat   end_lng
## 1     TA1307000163          13191  41.91824 -87.65711 41.92217 -87.63889
## 2            13277          13191  41.94011 -87.64545 41.92217 -87.63889
## 3     TA1306000015          13016  41.88592 -87.65113 41.89435 -87.62280
## 4     KA1503000038          13134  41.83846 -87.63541 41.88137 -87.67493
## 5            13247          13288  41.89595 -87.66773 41.92008 -87.67785
## 6     TA1309000010   KA1503000072  41.87068 -87.62571 41.88314 -87.63724
##   member_casual
## 1        member
## 2        casual
## 3        member
## 4        member
## 5        casual
## 6        member
```

```
str(data_01)
```

```
## 'data.frame':    181678 obs. of  11 variables:
##  $ ride_id         : chr  "65DBD2F447EC51C2" "0C201AA7EA0EA1AD" "E0B148CCB358A49D" "54C5775D2B7C9188" ...
##  $ rideable_type   : chr  "electric_bike" "classic_bike" "electric_bike" "classic_bike" ...
##  $ started_at      : chr  "2022-12-05 10:47:18" "2022-12-18 06:42:33" "2022-12-13 08:47:45" "2022-12-
##  $ ended_at        : chr  "2022-12-05 10:56:34" "2022-12-18 07:08:44" "2022-12-13 08:59:51" "2022-12-
##  $ start_station_id: chr  "TA1307000163" "13277" "TA1306000015" "KA1503000038" ...
##  $ end_station_id  : chr  "13191" "13191" "13016" "13134" ...
##  $ start_lat       : num  41.9 41.9 41.9 41.8 41.9 ...
##  $ start_lng       : num  -87.7 -87.6 -87.7 -87.6 -87.7 ...
##  $ end_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng         : num  -87.6 -87.6 -87.6 -87.7 -87.7 ...
##  $ member_casual   : chr  "member" "casual" "member" "member" ...
```

Export clean data into csv

```
# write.csv(data_01, "dataclean/202212-clean.csv", row.names = FALSE)
```