# Cleaning Cylistic Data 2023-03

2023-07-31

## Import data

```
data_01 <- read.csv(file="dataset/202303-divvy-tripdata.csv")
```

## Check data 01

Check the data type for each meta

```
str(data_01)
```

```
## 'data.frame':    258678 obs. of  13 variables:
##  $ ride_id           : chr  "6842AA605EE9FBB3" "F984267A75B99A8C" "FF7CF57CFE026D02" "6B61B916032CB6
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2023-03-16 08:20:34" "2023-03-04 14:07:06" "2023-03-31 12:28:09" "2023-
##  $ ended_at          : chr  "2023-03-16 08:22:52" "2023-03-04 14:15:31" "2023-03-31 12:38:47" "2023-
##  $ start_station_name: chr  "Clark St & Armitage Ave" "Public Rack - Kedzie Ave & Argyle St" "Orleans
##  $ start_station_id  : chr  "13146" "491" "620" "TA1306000003" ...
##  $ end_station_name  : chr  "Larrabee St & Webster Ave" "" "Clark St & Randolph St" "Sheffield Ave &
##  $ end_station_id    : chr  "13193" "" "TA1305000030" "13154" ...
##  $ start_lat         : num  41.9 42 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.7 -87.6 -87.6 -87.7 ...
##  $ end_lat           : num  41.9 42 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```
summary(data_01)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:258678      Length:258678      Length:258678      Length:258678
##  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:258678      Length:258678      Length:258678      Length:258678
##  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.   :41.63   Min.   :-87.85
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.89   Median :-87.64   Median :41.89   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.   :42.07   Max.   :-87.53   Max.   :42.08   Max.   :-87.52
##                                   NA's   :183     NA's   :183
##  member_casual
##  Length:258678
##  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[is.na(data_01$ride_id) | data_01$ride_id=="", ])
```

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

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

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

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

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

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

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

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

```
##       n
## 1 35910
```

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

```
##       n
## 1 35910
```

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

```
##       n
## 1 38438
```

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

```
##       n
## 1 38438
```

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

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

Missing value checking result :

ride_id: [0]

rideable_type: [0]

started_at: [0]

ended_at: [0]

start_station_name: [35,910]

start_station_id: [35,910]

end_station_name: [38,438]

end_station_id: [38,438]

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

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

Fill missing value result : empty data was replace with NA

### Check missing value data

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

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

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

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

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

```
##     n
## 1 183
```

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

```
##     n
## 1 183
```

Missing value checking result :

start latitude and langitude : [0]

end latitude and langitude : [183]

### Remove Missing value with NA

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

```
# 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
```

```
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

```
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 :  41.89904 42.07 41.6485NULL
```

```
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 :  -87.64784 -87.52823 -87.83NULL
```

```
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 :  41.89935 42.08 41.63NULL
```

```
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 :  -87.64804 -87.52 -87.85NULL
```

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

```
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 6842AA605EE9FBB3 electric_bike 2023-03-16 08:20:34 2023-03-16 08:22:52
## 2 F984267A75B99A8C electric_bike 2023-03-04 14:07:06 2023-03-04 14:15:31
## 3 FF7CF57CFE026D02  classic_bike 2023-03-31 12:28:09 2023-03-31 12:38:47
## 4 6B61B916032CB6D6  classic_bike 2023-03-22 14:09:08 2023-03-22 14:24:51
## 5 E55E61A5F1260040 electric_bike 2023-03-09 07:15:00 2023-03-09 07:26:00
## 6 123AAD676850F53C  classic_bike 2023-03-22 17:47:02 2023-03-22 18:01:29
##   start_station_id end_station_id start_lat start_lng  end_lat   end_lng
## 1            13146          13193  41.91841 -87.63645 41.92182 -87.64414
## 2              491           <NA>  41.97000 -87.71000 41.95000 -87.71000
## 3              620   TA1305000030  41.89820 -87.63754 41.88458 -87.63189
## 4    TA1306000003          13154  41.88872 -87.64445 41.91052 -87.65311
## 5            18067   TA1306000015  41.91448 -87.66801 41.88578 -87.65102
## 6              620   TA1309000061  41.89820 -87.63754 41.92914 -87.64908
##   member_casual
## 1        member
## 2        member
## 3        member
## 4        member
## 5        member
## 6        member
```

```r
str(data_01)
```

```
## 'data.frame':    258495 obs. of  11 variables:
##  $ ride_id         : chr  "6842AA605EE9FBB3" "F984267A75B99A8C" "FF7CF57CFE026D02" "6B61B916032CB6D6" ...
##  $ rideable_type   : chr  "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at      : chr  "2023-03-16 08:20:34" "2023-03-04 14:07:06" "2023-03-31 12:28:09" "2023-03-
##  $ ended_at        : chr  "2023-03-16 08:22:52" "2023-03-04 14:15:31" "2023-03-31 12:38:47" "2023-03-
##  $ start_station_id: chr  "13146" "491" "620" "TA1306000003" ...
##  $ end_station_id  : chr  "13193" NA "TA1305000030" "13154" ...
##  $ start_lat       : num  41.9 42 41.9 41.9 41.9 ...
##  $ start_lng       : num  -87.6 -87.7 -87.6 -87.6 -87.7 ...
##  $ end_lat         : num  41.9 42 41.9 41.9 41.9 ...
##  $ end_lng         : num  -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual   : chr  "member" "member" "member" "member" ...
```

## Export clean data into csv

```r
# write.csv(data_01, "dataclean/202303-clean.csv", row.names = FALSE)
```