Cleaning Cylistic Data 2023-06

2023-07-31

Import data

```
data_01 <- read.csv(file="dataset/202306-divvy-tripdata.csv")
```

Check data 01

Check the data type for each meta

```
str(data_01)
```

```
## 'data.frame':    719618 obs. of  13 variables:
##  $ ride_id           : chr  "6F1682AC40EB6F71" "622A1686D64948EB" "3C88859D926253B4" "EAD8A5E0259DEC
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2023-06-05 13:34:12" "2023-06-05 01:30:22" "2023-06-20 18:15:49" "2023-
##  $ ended_at          : chr  "2023-06-05 14:31:56" "2023-06-05 01:33:06" "2023-06-20 18:32:05" "2023-
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 42 42 42 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.9 42 42 ...
##  $ end_lng           : num  -87.7 -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:719618      Length:719618      Length:719618      Length:719618
##  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:719618      Length:719618      Length:719618      Length:719618
##  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.   :41.64   Min.   :-87.87   Min.   : 0.00   Min.   :-88.16
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64
##  Mean   :41.91   Mean   :-87.65   Mean   :41.91   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.   :42.11   Max.   :  0.00
##                                   NA's   :889     NA's   :889
##  member_casual
##  Length:719618
##  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 116259
```

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

```
##        n
## 1 116259
```

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

```
##        n
## 1 124050
```

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

```
##        n
## 1 124050
```

```
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: [116,259]

start_station_id: [116,259]

end_station_name: [124,050]

end_station_id: [124,050]

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
## 1 889
```

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

```
##     n
## 1 889
```

Missing value checking result :

start latitude and langitude : [0]

end latitude and langitude : [889]

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 :  41.90563 42.07 41.64NULL
```

```
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 :  -87.64743 -87.52823 -87.87NULL
```

```
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 :  41.90573 42.11 0NULL
```

```
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 :  -87.64731 0 -88.16NULL
```

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 6F1682AC40EB6F71 electric_bike 2023-06-05 13:34:12 2023-06-05 14:31:56
## 2 622A1686D64948EB electric_bike 2023-06-05 01:30:22 2023-06-05 01:33:06
## 3 3C88859D926253B4 electric_bike 2023-06-20 18:15:49 2023-06-20 18:32:05
## 4 EAD8A5E0259DEC88 electric_bike 2023-06-19 14:56:00 2023-06-19 15:00:35
## 5 5A36F21930D6A55C electric_bike 2023-06-19 15:03:34 2023-06-19 15:07:16
## 6 CF682EA7D0F961DB electric_bike 2023-06-09 21:30:25 2023-06-09 21:49:52
##   start_station_id end_station_id start_lat start_lng end_lat end_lng
## 1             <NA>           <NA>     41.91    -87.69   41.91  -87.70
## 2             <NA>           <NA>     41.94    -87.65   41.94  -87.65
## 3             <NA>           <NA>     41.95    -87.68   41.92  -87.63
## 4             <NA>           <NA>     41.99    -87.65   41.98  -87.66
## 5             <NA>           <NA>     41.98    -87.66   41.99  -87.65
## 6             <NA>           <NA>     41.99    -87.68   41.94  -87.65
##   member_casual
## 1        member
## 2        member
## 3        member
## 4        member
## 5        member
## 6        member
```

```r
str(data_01)
```

```
## 'data.frame':    718729 obs. of  11 variables:
##  $ ride_id         : chr  "6F1682AC40EB6F71" "622A1686D64948EB" "3C88859D926253B4" "EAD8A5E0259DEC88" ...
##  $ rideable_type   : chr  "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at      : chr  "2023-06-05 13:34:12" "2023-06-05 01:30:22" "2023-06-20 18:15:49" "2023-06-
##  $ ended_at        : chr  "2023-06-05 14:31:56" "2023-06-05 01:33:06" "2023-06-20 18:32:05" "2023-06-
##  $ start_station_id: chr  NA NA NA NA ...
##  $ end_station_id  : chr  NA NA NA NA ...
##  $ start_lat       : num  41.9 41.9 42 42 42 ...
##  $ start_lng       : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat         : num  41.9 41.9 41.9 42 42 ...
##  $ end_lng         : num  -87.7 -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/202306-clean.csv", row.names = FALSE)
```