Cleaning Cylistic Data 2022-11

2023-07-31

Import data

```
data_01 <- read.csv(file="dataset/202211-divvy-tripdata.csv")
```

Check data 01

Check the data type for each meta

```
str(data_01)
```

```
## 'data.frame':    337735 obs. of  13 variables:
##  $ ride_id           : chr  "BCC66FC6FAB27CC7" "772AB67E902C180F" "585EAD07FDEC0152" "91C4E7ED3C262F
##  $ rideable_type     : chr  "electric_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-11-10 06:21:55" "2022-11-04 07:31:55" "2022-11-21 17:20:29" "2022-
##  $ ended_at          : chr  "2022-11-10 06:31:27" "2022-11-04 07:46:25" "2022-11-21 17:34:36" "2022-
##  $ start_station_name: chr  "Canal St & Adams St" "Canal St & Adams St" "Indiana Ave & Roosevelt Rd"
##  $ start_station_id  : chr  "13011" "13011" "SL-005" "SL-005" ...
##  $ end_station_name  : chr  "St. Clair St & Erie St" "St. Clair St & Erie St" "St. Clair St & Erie S
##  $ end_station_id    : chr  "13016" "13016" "13016" "13016" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```
summary(data_01)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:337735      Length:337735      Length:337735      Length:337735
##  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:337735      Length:337735      Length:337735      Length:337735
##  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.   :-87.84   Min.   : 0.00   Min.   :-87.84
##  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 :-87.65
##  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.   :-87.52   Max.   :42.08   Max.   :  0.00
##                                   NA's   :230     NA's   :230
##  member_casual
##  Length:337735
##  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 51957
```

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

```
##       n
## 1 51957
```

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

```
##       n
## 1 54259
```

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

```
##       n
## 1 54259
```

```
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: [51,957] start_station_id: [51,957] end_station_name: [54,259] end_station_id: [54,259] 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
## 1 230
```

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

```
##     n
## 1 230
```

Missing value checking result :

start latitude and langitude : [0] end latitude and langitude : [230]

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 :  41.89928 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 :  -87.64812 -87.52 -87.84NULL
```

```
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 :  41.89856 42.08 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 :  -87.64625 0 -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

```
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 BCC66FC6FAB27CC7 electric_bike 2022-11-10 06:21:55 2022-11-10 06:31:27
## 2 772AB67E902C180F  classic_bike 2022-11-04 07:31:55 2022-11-04 07:46:25
## 3 585EAD07FDEC0152  classic_bike 2022-11-21 17:20:29 2022-11-21 17:34:36
## 4 91C4E7ED3C262FF9  classic_bike 2022-11-25 17:29:34 2022-11-25 17:45:15
## 5 709206A3104CABC8  classic_bike 2022-11-29 17:24:25 2022-11-29 17:42:51
## 6 11DE62E16D1A6BD1  classic_bike 2022-11-04 14:40:47 2022-11-04 14:52:35
##   start_station_id end_station_id start_lat start_lng  end_lat   end_lng
## 1            13011          13016  41.87940 -87.63985 41.89435 -87.62280
## 2            13011          13016  41.87926 -87.63990 41.89435 -87.62280
## 3           SL-005          13016  41.86789 -87.62304 41.89435 -87.62280
## 4           SL-005          13016  41.86789 -87.62304 41.89435 -87.62280
## 5           SL-005          13016  41.86789 -87.62304 41.89435 -87.62280
## 6            13022   TA1306000003  41.89228 -87.61204 41.88872 -87.64445
##   member_casual
## 1        member
## 2        member
## 3        member
## 4        member
## 5        member
## 6        member
```

```
str(data_01)
```

```
## ’data.frame’:    337505 obs. of  11 variables:
##  $ ride_id         : chr  "BCC66FC6FAB27CC7" "772AB67E902C180F" "585EAD07FDEC0152" "91C4E7ED3C262FF9
##  $ rideable_type   : chr  "electric_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at      : chr  "2022-11-10 06:21:55" "2022-11-04 07:31:55" "2022-11-21 17:20:29" "2022-11
##  $ ended_at        : chr  "2022-11-10 06:31:27" "2022-11-04 07:46:25" "2022-11-21 17:34:36" "2022-11
##  $ start_station_id: chr  "13011" "13011" "SL-005" "SL-005" ...
##  $ end_station_id  : chr  "13016" "13016" "13016" "13016" ...
##  $ start_lat       : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng       : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng         : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual   : chr  "member" "member" "member" "member" ...
```

Export clean data into csv

```
# write.csv(data_01, "dataclean/202211-clean.csv", row.names = FALSE)
```