Data Manipulations and Sources

To start with, I downloaded this data from the Chicago Data Portal. I used the CTA - Ridership - 'L' Station Entries - Daily Totals link in particular. I first exported the data as a tsv file, a tab delimited value list. As of 2/14/22, this file contained over one million entries. These entries contained the station_id, a unique identifiable number the differentiate different stations across the L Stations, it also contained the station name, the date that the entry was logged as, the daytype, and number of rides that particular station had on that day. The daytype is logged differently than the normal Mon, Tue, Wed,etc. that we are used to for days of the week. They log it as W - weekday, A-Saturday, U - Sunday and specific holidays like New Year's Day, Memorial Day, Independence Day, Labor Day, Thanksgiving, and Christmas day. All other holidays are recorded as the day they fall on.

Once I had this file downloaded, I opened it into Notepad++ and sorted it lexicographically so the entries in the file are sorted based on the Station Name.

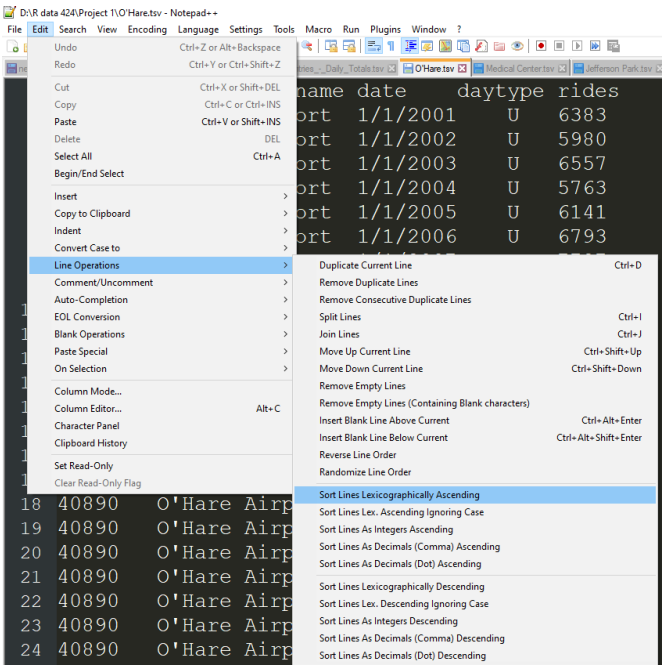


Figure 1: Sorting the tsv

From this point on, I selected the UIC-Halsted station entries, the O'Hare entries, and Jefferson Park entries and pasted each of them into their own separate files.

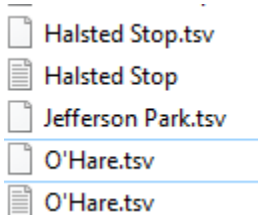


Figure 2: File separation

The next thing I did inside R, was to read in each file with the read.table() function. Now that the data's in a usable format we can begin manipulating it. First we convert the date column into a usable date object in R using the as.Date() function so now instead of the date looking like mm/dd/yyyy, it now looks like yyyy-mm-dd.

```
#converting date type to workable column
newDate <- as.Date(Halsted$date, "%m/%d/%Y")
Halsted$newDate<-newDate
Halsted$date<-NULL
```

Figure 3: Date manipulation

Other types of data manipulation that we do is different types of summation. First, for each year, we add up all the rides that comprise the year's data and then we graph it. Another kind data manipulation that I did is I grouped all the data by months and used that to calculate the rides that were taken on a specific station by that month.

```
output$eachMonth <- renderPlot({
  subset(Halsted,newDate > as.Date("2020-12-31")) %>%
    ggplot(aes(month(newDate,label = TRUE),rides))+
    geom_bar(stat = "identity",fill="#88CCEE")+
    labs(x = "Months", y = "Number of Entries", title = "Entries per Month")+
    theme_bw()+
    scale_y_continuous(expand = c(0,0))
```

Figure 4: Month Entries Code

The next data manipulation that I did was I grouped the days in 2021 and then grouped by the week day that the date fell on and then summed those rides together. We needed to do this because the daytype column in the original data wasn't specific enough for us, a W to indicate week day was too general when we needed to use wday() function in order to grab the day of the week that a specific date fell on.

```
#groups sum of Rides per day together
temp1 = subset(Halsted,newDate > as.Date("2020-12-31"))
sumOfRidesPerDay = temp1 %>% group_by(wday(newDate)) %>% summarise(sum = sum(rides))
```

Figure 5: Week Day Entries Code