

# Querying Chicago Rideshare Data

*Enoch Shin, Felix Stetsenko*

*December 07, 2019*

## Purpose

This file ingests the transportation network provider (TNP) data from the City of Chicago data portal.

## Query City of Chicago Data

City of Chicago data uses the Socrata API. You'll have to get an API key from the Chicago Data Portal.

Navigate to the Socrata developer page (<https://dev.socrata.com/foundry/data.cityofchicago.org/m6dm-c72p>) for the Chicago data and scroll down. Click the button “sign up for an app token,” and make an account to do so.

Once you get your key, you can navigate to the appropriate data page on the City of Chicago Data Portal (<https://data.cityofchicago.org/Transportation/Transportation-Network-Providers-Trips/m6dm-c72p>). Click the “API” button in the top right corner and copy the `.json` link.

```
url <- "https://data.cityofchicago.org/resource/m6dm-c72p.json"
mytoken <- "XXXXXXXXXXXXXXXXXX"
```

We want to query Valentine's Day 2019 (February 14, 2019): the query is commented out below. Do some wrangling to make sure the desired variables are coded correctly.

```
#####QUERY#####
# valday <- read.socrata(paste0(url, "?",
#"$where=trip_start_timestamp between '2019-02-14T00:00:00' and '2019-02-14T23:59:59'"),
#app_token=mytoken)

# saveRDS(valday, "ChicagoCommute/RDA/valday.Rda")
valday <- readRDS("ChicagoCommute/RDA/valday.Rda")

# remove the coordinate variable, which is a list
valday <- valday %>% mutate_at(c("pickup_centroid_latitude",
                                "pickup_centroid_longitude",
                                "dropoff_centroid_longitude",
                                "dropoff_centroid_latitude"), as.numeric) %>%
  dplyr::select(-contains(".coordinate"))
# write to feather for both the app folder and feather folder
write_feather(valday, "ChicagoCommute/feather/valday.feather")
write_feather(valday, "ChicagoCommute/app/feather/valday.feather")
```

## Univariate Glimpses into Data

```
nrow(valday)
```

```
## [1] 315613
```

```
mosaic::favstats(valday$trip_miles)
```

```
## Registered S3 method overwritten by 'mosaic':
```

```
##   method          from
```

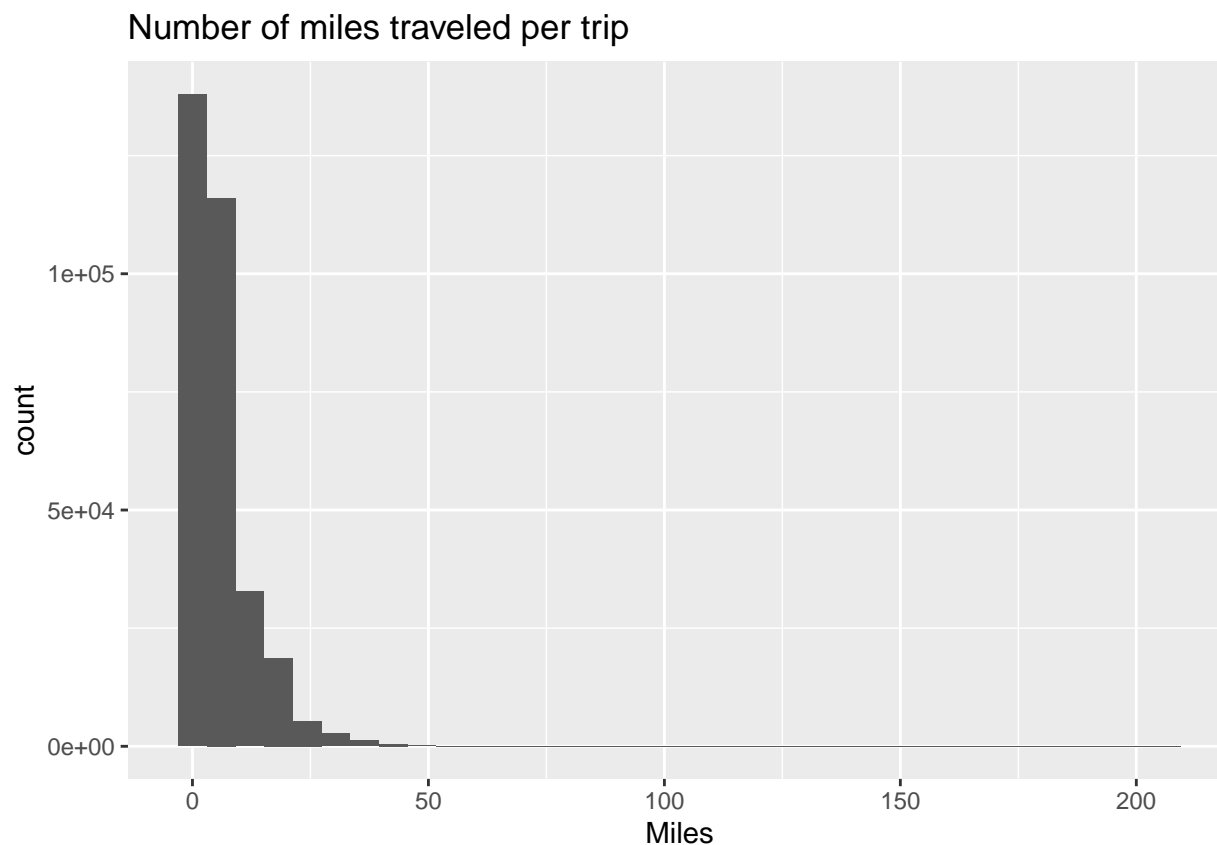
```
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##   min      Q1   median      Q3    max    mean      sd      n missing
##     0 1.750257 3.603295 7.453934 206.481 5.900087 6.465699 315613      0
```

```
mosaic::favstats(valday$trip_seconds)/60
```

```
##           min  Q1   median      Q3 max    mean      sd      n missing
##    0.03333333 9.25 15.31667 24.91667 868 19.04572 14.15609 5260.217      0
```

```
ggplot(valday, aes(x=as.numeric(trip_miles))) +  
  geom_histogram(bins=35) +  
  ggtitle("Number of miles traveled per trip") +  
  xlab("Miles")
```



```
# count trips based on before or after sunset
# sunset on Valentine's Day 2019 was at 5:22 CST
## which is 11:22 PM UTC
sunset <- lubridate::as_datetime("2019-02-14 23:22:00")
sunset
```

```
## [1] "2019-02-14 23:22:00 UTC"
```

```
day <- filter(valday, valday$trip_start_timestamp < sunset)
evening <- filter(valday, valday$trip_start_timestamp >= sunset)

nrow(day)/nrow(valday)
```

```
## [1] 0.6561707
```

```
nrow(evening)/nrow(valday)
```

```
## [1] 0.3438293
```

## Sample the data for smaller set

The data in its raw form is too large to plot with leaflet. So, we'll take a random sample of n=500.

```
valday <- read_feather("ChicagoCommute/feather/valday.feather")
set.seed(32)
valday_small <- valday[sample(nrow(valday), 500),]

write_feather(valday_small, "ChicagoCommute/feather/valday_small.feather")
write_feather(valday_small, "ChicagoCommute/app/feather/valday_small.feather")
```

## Appendix: experiments in visualization

### Making Simple Feature Object for Points

Our main visualization uses polygons to draw lines between pickups and dropoffs. This is useful since we get an idea of where people are going for each trip. I initially tried to make simple feature objects instead of making polygons in leaflet, but this proved to be unnecessary and inferior since the `st_as_sf` function cannot handle NA values.

```
valday <- read_feather("ChicagoCommute/feather/valday.feather")

pickups <- valday %>%
  drop_na("pickup_centroid_longitude") %>%
  drop_na("pickup_centroid_latitude")

dropoffs <- valday %>%
  drop_na("dropoff_centroid_longitude") %>%
  drop_na("dropoff_centroid_latitude")

pts_pickup <- st_as_sf(pickups, coords = c("pickup_centroid_longitude",
                                           "pickup_centroid_latitude"),
                      crs = 4326, na.fail = FALSE)

pts_dropoff <- st_as_sf(dropoffs, coords = c("dropoff_centroid_longitude",
                                             "dropoff_centroid_latitude"),
                       crs = 4326, na.fail = FALSE)

# feather cannot write a list object, so use RDS
saveRDS(pts_pickup, "ChicagoCommute/RDA/pts_pickup.Rda")
saveRDS(pts_pickup, "ChicagoCommute/app/RDA/pts_pickup.Rda")

saveRDS(pts_dropoff, "ChicagoCommute/RDA/pts_dropoff.Rda")
saveRDS(pts_dropoff, "ChicagoCommute/app/RDA/pts_dropoff.Rda")
```