R Notebook

Code

Hide

```
library(corrplot)
```

```
corrplot 0.94 loaded
```

Hide

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.4     ✓ readr     2.1.5
✓ forcats   1.0.0     ✓ stringr   1.5.1
✓ ggplot2   3.5.1     ✓ tibble    3.2.1
✓ lubridate 1.9.3     ✓ tidyr     1.3.1
✓ purrr     1.0.2
── Conflicts ────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors
```

Hide

```
library(ggplot2)
library(maps)
```

```


Attaching package: ‘maps’

The following object is masked from ‘package:purrr’:

    map
```

Hide

```
library(ggmap)
```

```
ℹ Google's Terms of Service: ]8;;https://mapsplatform.google.com<https://mapsplatform.google.com>]8;;
  Stadia Maps' Terms of Service: ]8;;https://stadiamaps.com/terms-of-service/<https://stadiamaps.com/terms-of-service/>]8;;
  OpenStreetMap's Tile Usage Policy: ]8;;https://operations.osmfoundation.org/policies/tiles/<https://operations.osmfoundation.org/policies/tiles/>]8;;
ℹ Please cite ggmap if you use it! Use `citation("ggmap")` for details.
```

Hide

```
library(ggplot2)
library(tmap)
```

```
Registered S3 method overwritten by 'htmlwidgets':
  method           from
  print.htmlwidget tools:rstudio
Breaking News: tmap 3.x is retiring. Please test v4, e.g. with
remotes::install_github('r-tmap/tmap')
```

Hide

```
library(geosphere)
library(sf)
```

```
Linking to GEOS 3.12.1, GDAL 3.8.4, PROJ 9.3.1; sf_use_s2() is TRUE
```

Hide

```
trips_df <- read_csv("../data/trips_data.csv")
```

```
Rows: 6453999 Columns: 13
── Column specification ────────────────────────────────────────────────────────────────────────────────────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#exploring the data

Hide

```
head(trips_df)
```

ride_id <chr>	rideable_type <chr>	started_at <S3: POSIXct>	ended_at <S3: POSIXct>	start_station_name <chr>
6F1682AC40EB6F71	electric_bike	2023-06-05 13:34:12	2023-06-05 14:31:56	*NA*
622A1686D64948EB	electric_bike	2023-06-05 01:30:22	2023-06-05 01:33:06	*NA*
3C88859D926253B4	electric_bike	2023-06-20 18:15:49	2023-06-20 18:32:05	*NA*
EAD8A5E0259DEC88	electric_bike	2023-06-19 14:56:00	2023-06-19 15:00:35	*NA*
5A36F21930D6A55C	electric_bike	2023-06-19 15:03:34	2023-06-19 15:07:16	*NA*
CF682EA7D0F961DB	electric_bike	2023-06-09 21:30:25	2023-06-09 21:49:52	*NA*

6 rows | 1-5 of 13 columns

Hide

```
tail(trips_df)
```

ride_id <chr>	rideable_type <chr>	started_at <S3: POSIXct>	ended_at <S3: POSIXct>
E3BEED04143797AC	electric_bike	2024-06-24 17:12:04	2024-06-24 17:26:15
1D1EBE57758FB1EE	electric_bike	2024-06-11 08:25:42	2024-06-11 08:33:43
2F63E9CD01D79515	electric_bike	2024-06-24 11:40:44	2024-06-24 11:42:09
97D225818F9C7AC3	electric_bike	2024-06-30 10:43:32	2024-06-30 10:45:45
C8D2A48B901F7399	electric_bike	2024-06-11 18:20:40	2024-06-11 18:29:04
C372E7A1A7BA19D4	electric_bike	2024-06-15 15:48:49	2024-06-15 15:52:31

6 rows | 1-4 of 13 columns

Hide

```
cat("Dimensions:\n\n")
```

```
Dimensions:
```

Hide

```
dim(trips_df)
```

```
[1] 6453999      13
```

Hide

```
cat("\nsummary:\n\n")
```

```

summary:
```

Hide

```
summary(trips_df)
```

```
  ride_id          rideable_type        started_at
Length:6453999     Length:6453999     Min.   :2023-06-01 00:00:44.00
Class :character   Class :character   1st Qu.:2023-08-05 15:30:47.00
Mode  :character   Mode  :character   Median :2023-10-17 07:39:52.00
                                      Mean   :2023-11-26 07:45:49.20
                                      3rd Qu.:2024-04-12 14:29:54.00
                                      Max.   :2024-06-30 23:55:17.06

   ended_at                      start_station_name start_station_id
Min.   :2023-06-01 00:02:56.00   Length:6453999     Length:6453999
1st Qu.:2023-08-05 15:53:51.00   Class :character   Class :character
Median :2023-10-17 07:50:36.00   Mode  :character   Mode  :character
Mean   :2023-11-26 08:04:19.18
3rd Qu.:2024-04-12 14:45:59.50
Max.   :2024-06-30 23:59:57.93

end_station_name   end_station_id       start_lat       start_lng
Length:6453999     Length:6453999     Min.   :41.63   Min.   :-87.94
Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
Mode  :character   Mode  :character   Median :41.90   Median :-87.64
                                      Mean   :41.90   Mean   :-87.65
                                      3rd Qu.:41.93   3rd Qu.:-87.63
                                      Max.   :42.07   Max.   :-87.46

   end_lat         end_lng       member_casual
Min.   : 0.00   Min.   :-88.16   Length:6453999
1st Qu.:41.88   1st Qu.:-87.66   Class :character
Median :41.90   Median :-87.64   Mode  :character
Mean   :41.90   Mean   :-87.65
3rd Qu.:41.93   3rd Qu.:-87.63
Max.   :42.19   Max.   :  0.00
NA's   :8808    NA's   :8808
```

there seems to be missing values on end_lat, and end_lng. There are 6 categorical columns, rideable_type, and menber_casual, start_station_name/id, and end_station_name/id. There also seems to be an issue with end_lat were it has a minimum 0.00 and end_lng with a max of 0.00 this doesn't make sense since Chicago is around the values lattitude: 41.739685, longitude : -87.554420.

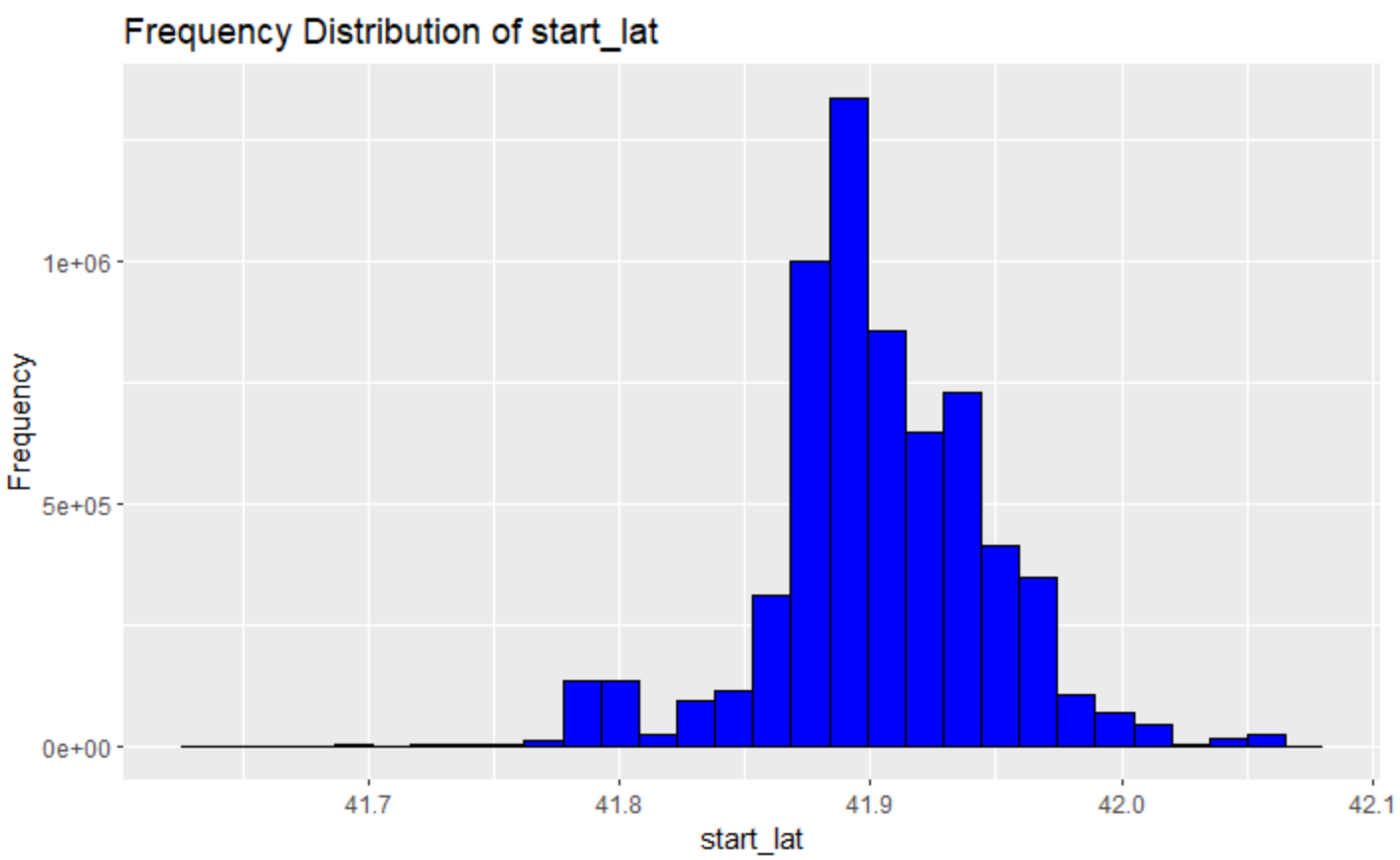
##checking spread of the data

longitude and latitude data.

Hide

```
ggplot(trips_df, aes(x = start_lat)) +
    geom_histogram(fill = "blue", color = "black") +
    labs(title = "Frequency Distribution of start_lat", x = "start_lat", y = "Frequency")
```

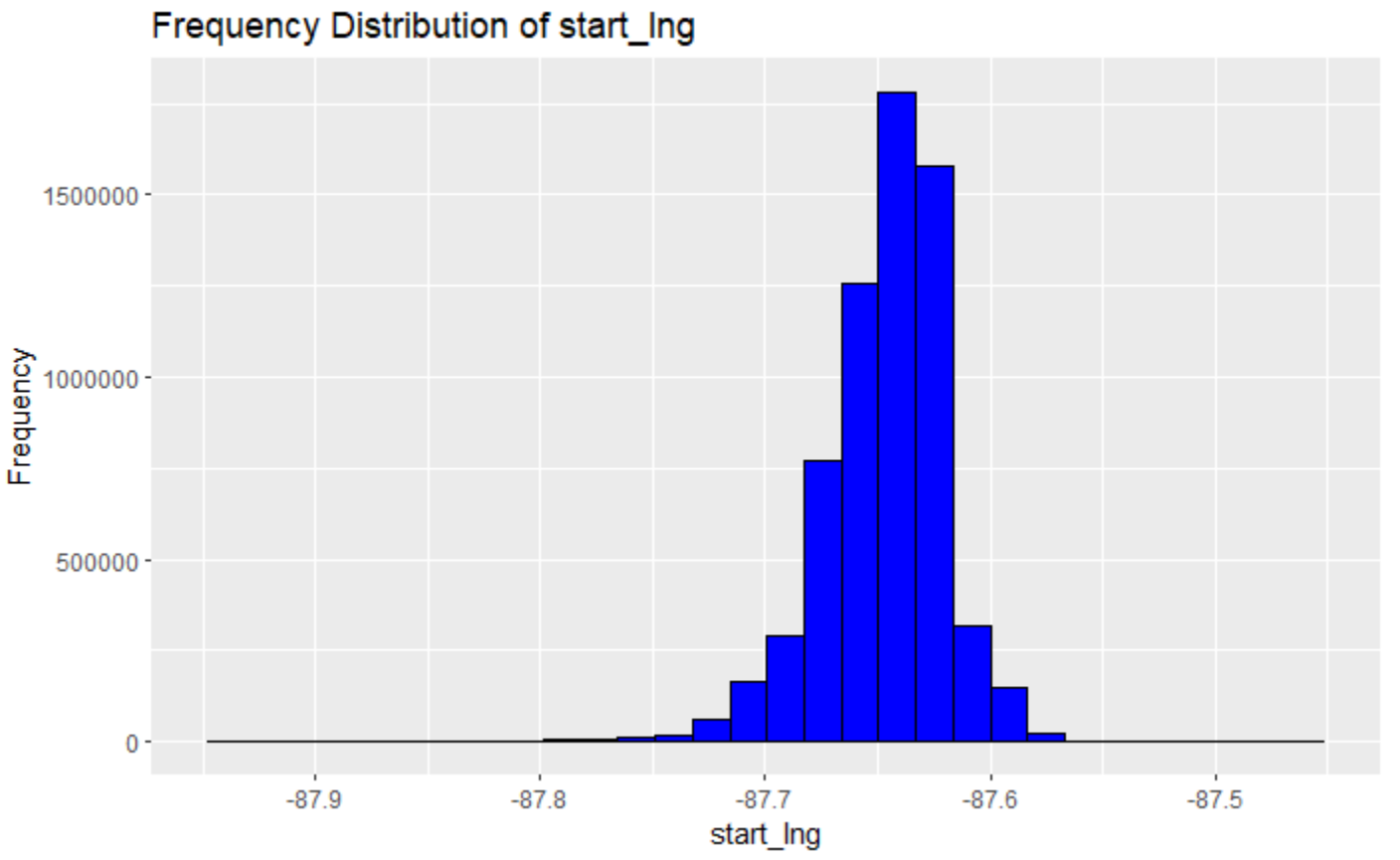
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Hide

```
ggplot(trips_df, aes(x = start_lng)) +
    geom_histogram(fill = "blue", color = "black") +
    labs(title = "Frequency Distribution of start_lng", x = "start_lng", y = "Frequency")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

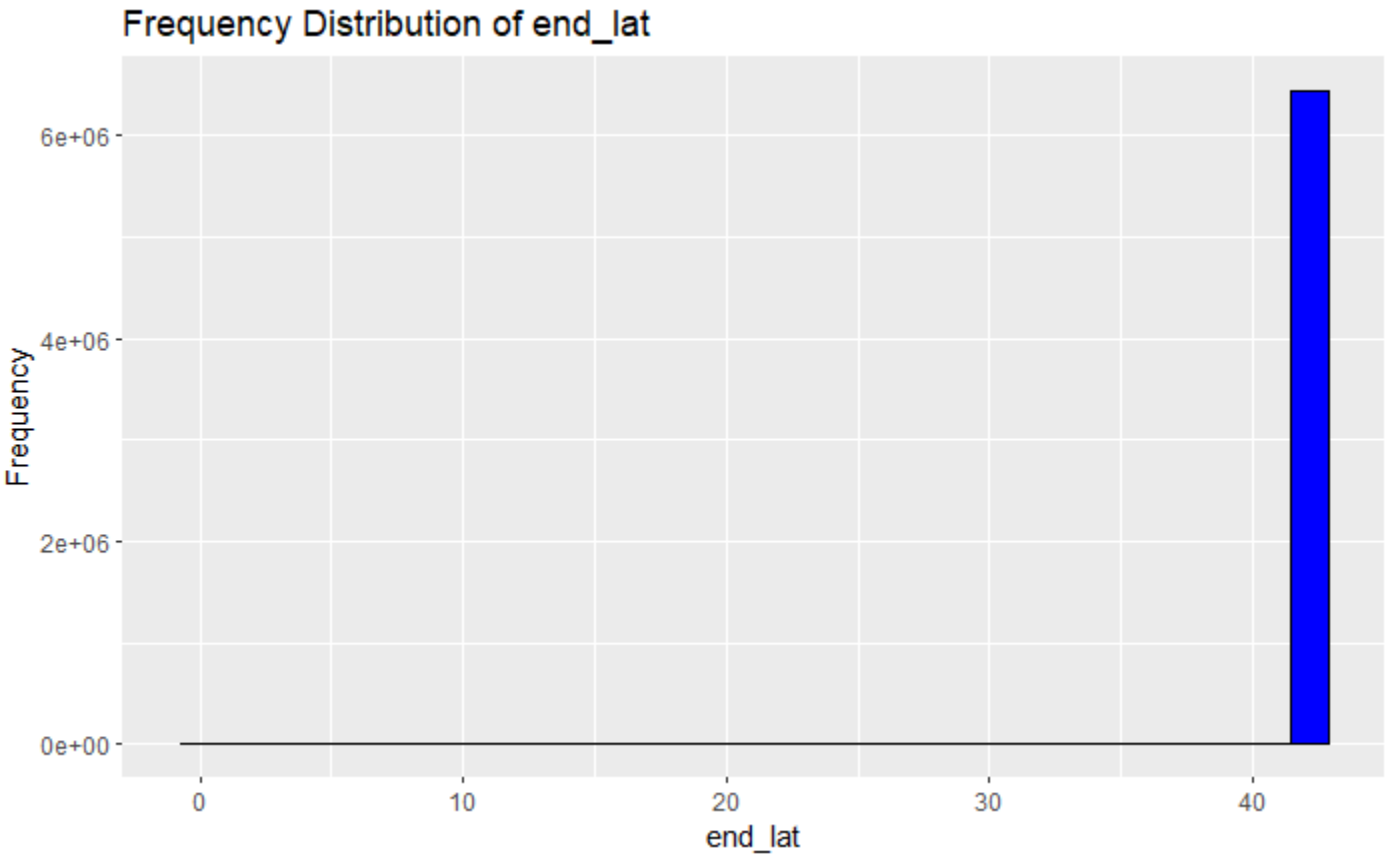


Hide

```
ggplot(trips_df, aes(x = end_lat)) +
    geom_histogram(fill = "blue", color = "black") +
    labs(title = "Frequency Distribution of end_lat", x = "end_lat", y = "Frequency")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
Warning: Removed 8808 rows containing non-finite outside the scale range (`stat_bin()`).
```

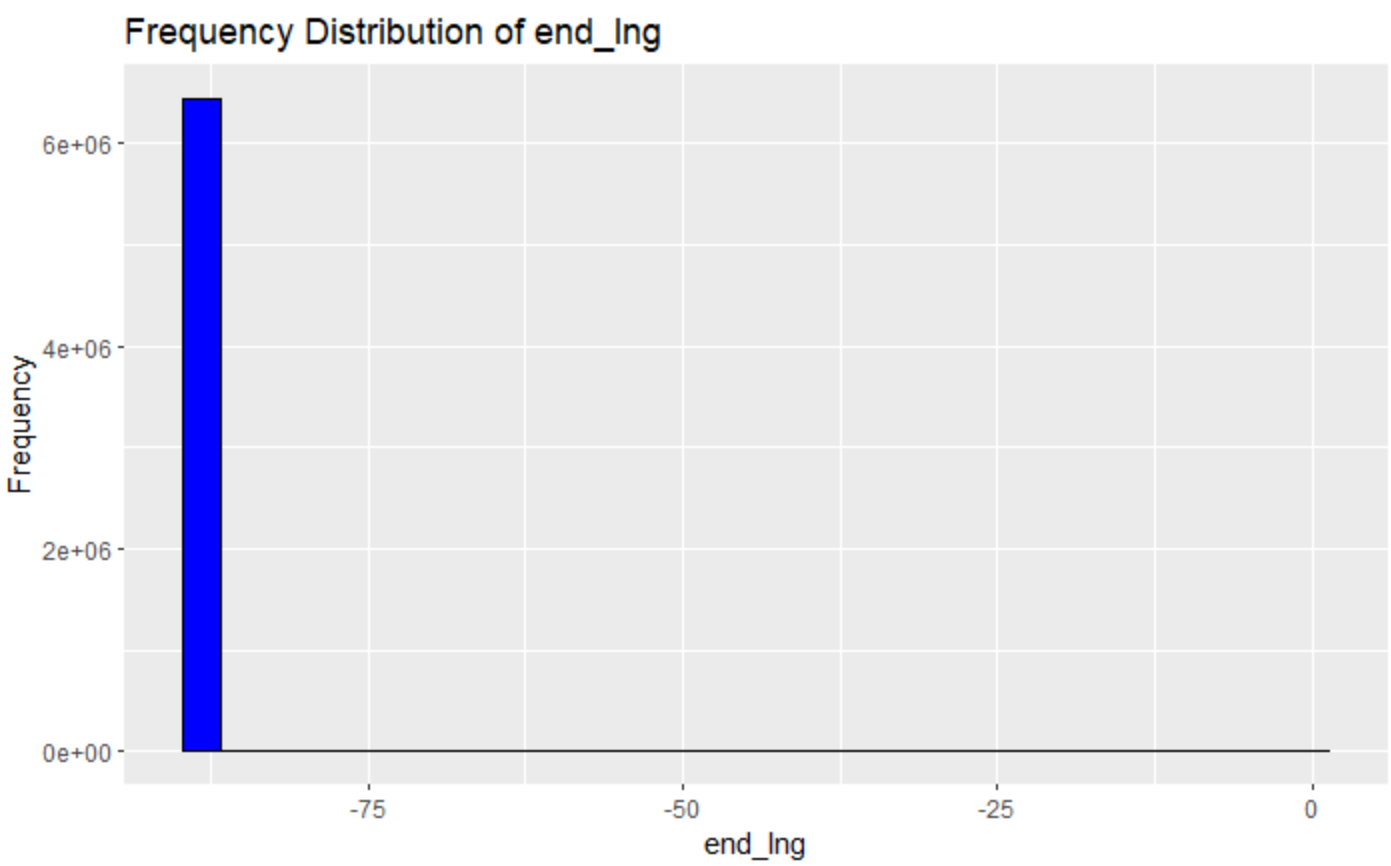


Hide

```
ggplot(trips_df, aes(x = end_lng)) +
    geom_histogram(fill = "blue", color = "black") +
    labs(title = "Frequency Distribution of end_lng", x = "end_lng", y = "Frequency")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
Warning: Removed 8808 rows containing non-finite outside the scale range (`stat_bin()`).
```



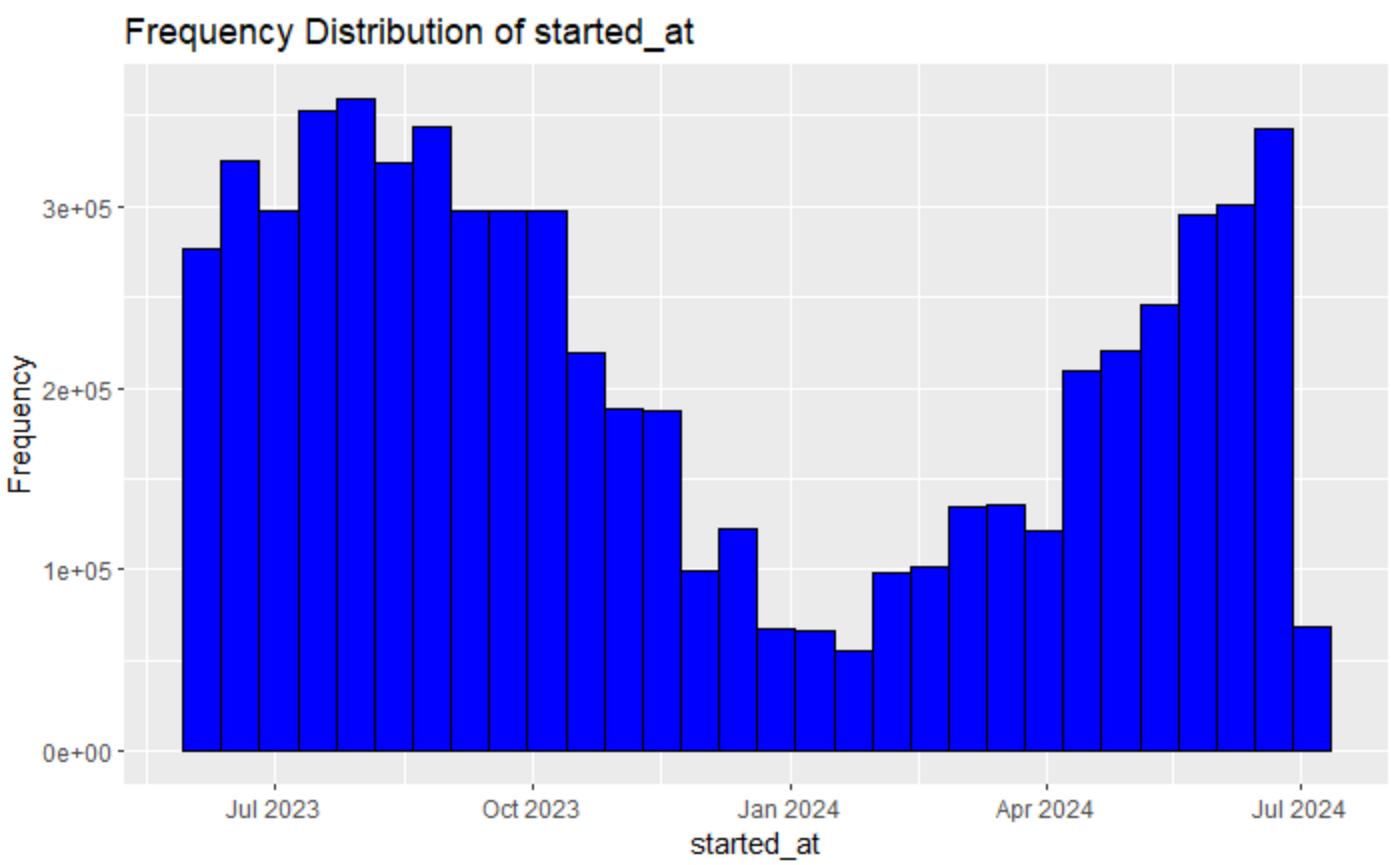
end_lng and end_lat seem to have outliers on them so That is another problem to consider.

checking started_at and ended_at.

Hide

```
ggplot(trips_df, aes(x = started_at)) +
    geom_histogram(fill = "blue", color = "black") +
    labs(title = "Frequency Distribution of started_at", x = "started_at", y = "Frequency")
```

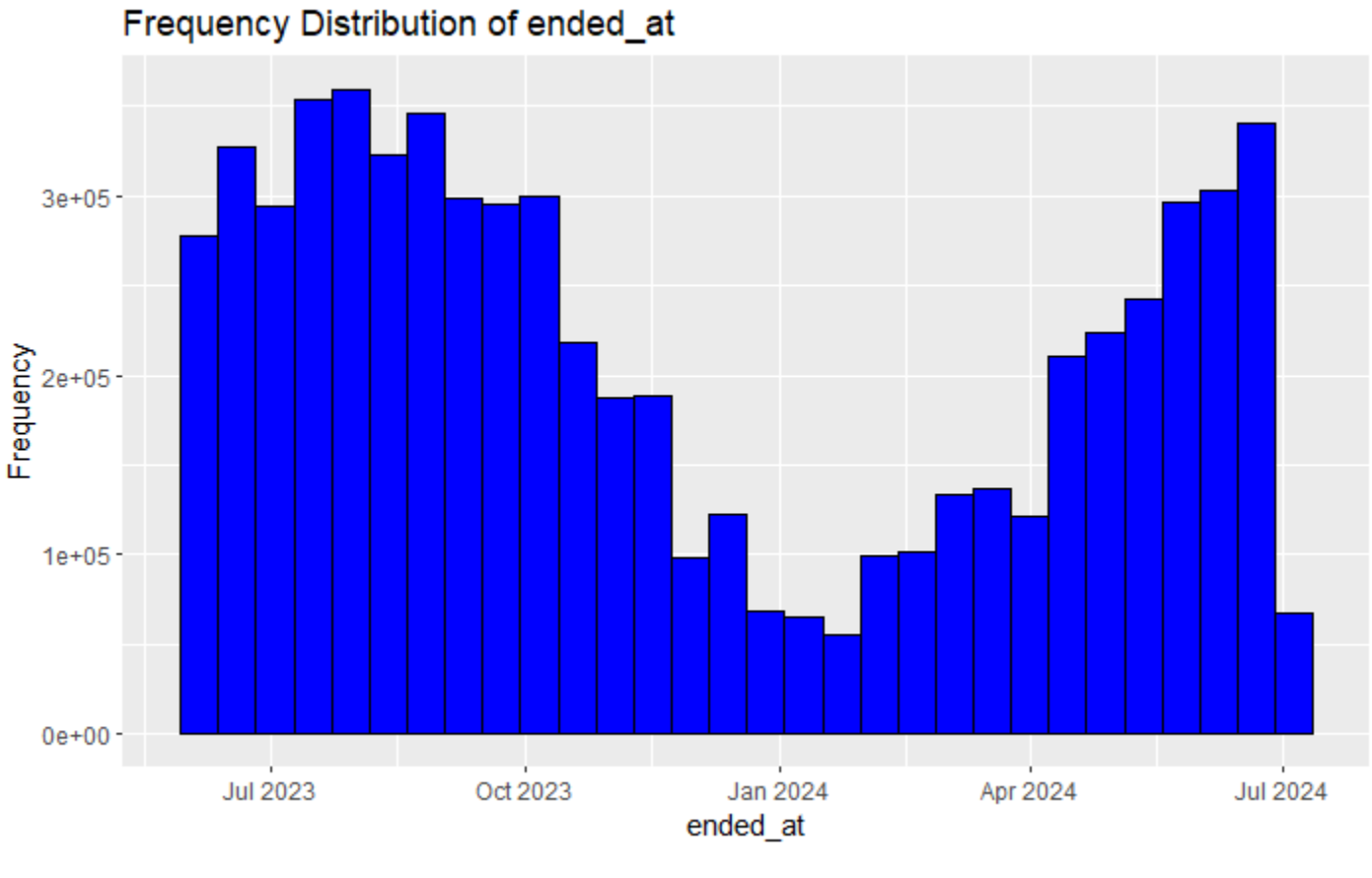
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Hide

```
ggplot(trips_df, aes(x = ended_at)) +
    geom_histogram(fill = "blue", color = "black") +
    labs(title = "Frequency Distribution of ended_at", x = "ended_at", y = "Frequency")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

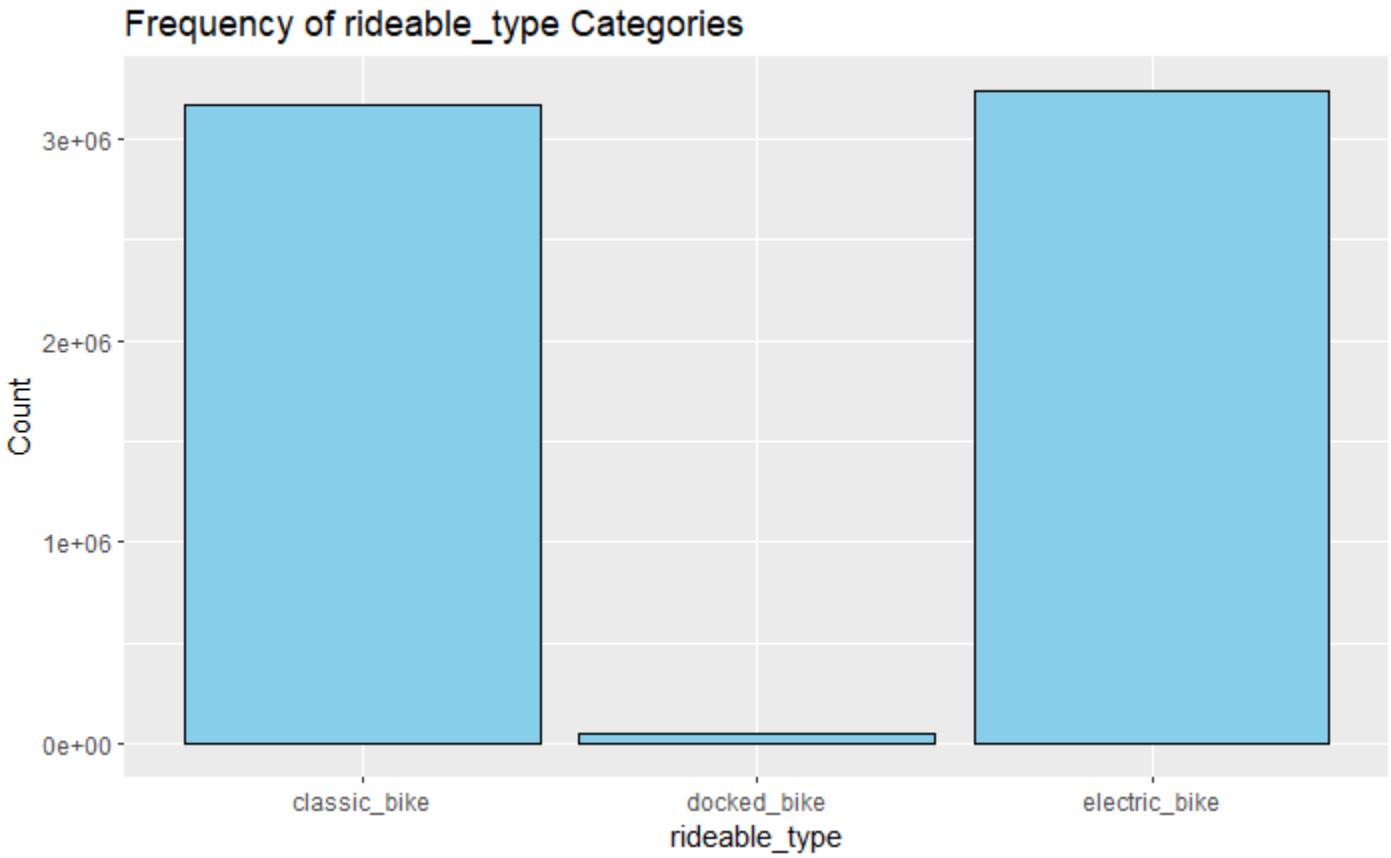


This all seems in order less trips on january make sense since its very cold and both have around the same spread. As if a trip started it will end.

checking ridable_type and menber_casual.

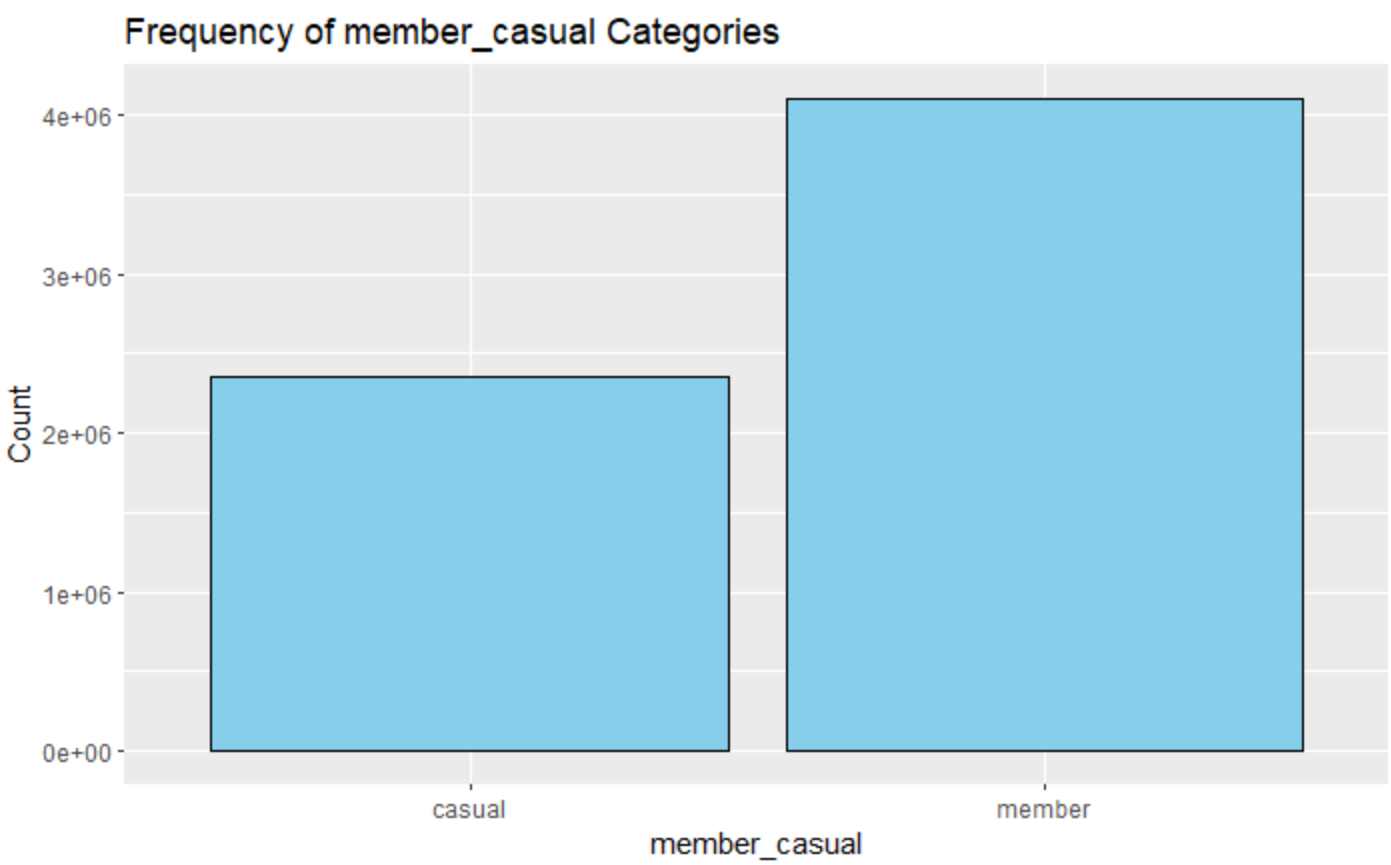
Hide

```
ggplot(trips_df, aes(x = rideable_type)) +
    geom_bar(fill = "skyblue", color = "black") +
    labs(title = "Frequency of rideable_type Categories", x = "rideable_type", y = "Count")
```



Hide

```
ggplot(trips_df, aes(x = member_casual)) +
    geom_bar(fill = "skyblue", color = "black") +
    labs(title = "Frequency of member_casual Categories", x = "member_casual", y = "Count")
```



There is a docked_bike. This is weird since when looking at the types of bike that divvy data gives. There is only classic and electric. This also seems to be a very small portion of the data.

Hide

```
trips_df %>%
  select(everything()) %>%
  filter(rideable_type == "docked_bike")
```

ride_id <chr>	rideable_type <chr>	started_at <S3: POSIXct>	ended_at <S3: POSIXct>
3CC49A8C761A669B	docked_bike	2023-06-09 21:54:25	2023-06-10 06:31:43
928BC74967190966	docked_bike	2023-06-18 12:16:43	2023-06-18 12:46:23
C7A10EF1C29DFEDC	docked_bike	2023-06-19 09:03:56	2023-06-19 09:06:17
AB91410999F7DB52	docked_bike	2023-06-20 19:57:51	2023-06-20 21:40:08
8B1E1CCA45C2B452	docked_bike	2023-06-23 09:43:16	2023-06-23 10:29:53
7E1B325BC701385F	docked_bike	2023-06-22 13:37:55	2023-06-22 13:42:06
24C9980F643D3829	docked_bike	2023-06-04 19:59:53	2023-06-04 20:21:56
A5ED464909854EE0	docked_bike	2023-06-27 12:53:59	2023-06-27 18:17:50
7192B46F15A82E5C	docked_bike	2023-06-04 19:59:41	2023-06-04 20:21:50
5EB3B94A07F5A71E	docked_bike	2023-06-15 09:28:45	2023-06-15 09:38:52

1-10 of 49,355 rows | 1-4 of 13 columns Previous **1** 2 3 4 5 6 … 100 Next

From what I could find out online this are supposed to be classic trips. However I don't want to just change them in case there is something important about them.

Are all ids unique?

Hide

```
trips_df %>%
  filter(duplicated(.)) %>%
  count()
```

n
<int>
0

1 row

yes, they are all unique.

##checking how much data is actually missing

Before we saw that start_station_name, start_station_id, end_station_name and end_station_id, end_lng and end_lat are missing. I want to make sure I didn't miss anything before proceding to the cleaning.

Hide

```
trips_df %>%
  select(everything()) %>%
  summarise(across(everything(), ~sum(is.na(.))))
```

ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_nam
<int>	<int>	<int>	<int>	<int>	<int>	<int
0	0	0	0	1049262	1049262	110460

1 row | 1-7 of 13 columns

There doesn't seem to be any more columns missing. It also looks like ids and names are always missing at the same time. The same seems to be the case for end_lat and end_lng.

##How is the missing data related?

In the code bellow I am looking to see if there are cases where start_station_name and id are not missing at the same time. Then I am doing the same with end_station_name/id and end_lng/lat

Hide

```
trips_df %>%
  select(start_station_id,end_station_id, start_station_name, end_station_name,end_lat,end_lng) %>%
  filter(
    (!complete.cases(start_station_name) & complete.cases(start_station_id))
    | (!complete.cases(end_station_name) & complete.cases(end_station_id))
    | (!complete.cases(end_lng) & complete.cases(end_lat))) %>%
  count()
```

n
<int>
0

1 row

With this I know that both start ids and start names are always missing at the same time as well as end ids are always missing at the same time as end names. Longitude and lattitudes are also always missing at the same time.

With all of this information we know that there is are 3 chunks of data missing. One being end_lng and end_lat. The other one being start_station_id and start_station_name, and finally Finally we got end_station_id, and end_station_name.