Cyclist Case Study on Google Data Analytics Course

Akila Ekanayake

11/18/2021

Setting up with using libraries

Case Study in brief

Cyclistic offers a bike-sharing service since 2016. “In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime.”

Stakeholders wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights,they will design a new marketing strategy to convert casual riders into annual members.

Importing the Data

With the onlinr version of R-Studio I couldn’t extract and load all the csv files at once. With the Desktop version I have used a method that have presented in another GitHub repository (https://github.com/10below/arpeggio (https://github.com/10below/arpeggio)).

From this method I have extracted the csv files into a single folder that have defined as my workplace. Below code snippet have used to read and concat all the csv files into a single dataframe.

```
## Adding the .csv files to file list. Since I have saved the files in the working directory pat
h does not have to defined. In case you used another place you have to provide the path correctl
y
file_list <- list.files(path = ".", pattern = "*-divvy-tripdata.csv")

## Extracting the csv file list to "data" object
data <- ldply(.data = file_list, .fun=read.csv)
```

Dataset overlook

Data set have 5,378,834 observations

```
str(data)
```

```
## 'data.frame':    84776 obs. of  13 variables:
##  $ ride_id           : chr  "A847FADBBC638E45" "5405B80E996FF60D" "5DD24A79A4E006F4" "2A59BBD
F5CDBA725" ...
##  $ rideable_type     : chr  "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : chr  "2020-04-26 17:45:14" "2020-04-17 17:08:54" "2020-04-01 17:54:13"
"2020-04-07 12:50:19" ...
##  $ ended_at          : chr  "2020-04-26 18:12:03" "2020-04-17 17:17:03" "2020-04-01 18:08:36"
"2020-04-07 13:02:31" ...
##  $ start_station_name: chr  "Eckhart Park" "Drake Ave & Fullerton Ave" "McClurg Ct & Erie St"
"California Ave & Division St" ...
##  $ start_station_id  : int  86 503 142 216 125 173 35 434 627 377 ...
##  $ end_station_name  : chr  "Lincoln Ave & Diversey Pkwy" "Kosciuszko Park" "Indiana Ave & Ro
osevelt Rd" "Wood St & Augusta Blvd" ...
##  $ end_station_id    : int  152 499 255 657 323 35 635 382 359 508 ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 42 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

Data cleaning and Preprocessing

###Original data in the dataframe ride_id : A unique identifier for the trip rideable_type : The type of bicyle (Classic, Docked, Electric) started_at : Datetime of when the trip started ended_at : Datetime of when the trip ended start_station_name : The name of the dock station where the trip started start_station_id : The ID of the dock station where the trip started end_station_name, end_station_id : Same for where the trip ended start_lat, start_lng : The GPS coordinates of the trip start end_lat, end_lng : The GPS coordinates of the trip end member_casual : Membership type (casual or member)

Changing “started_at” and “ended_at” data types into datetime

Originally starting dates and ending dates were recognized as “char”. They had to change into datetime data type for calculations

```
data$started_at <- anytime(data$started_at)
data$ended_at <- anytime(data$ended_at)
```

Adding new columns to the dataframe

By adding new columns we can filter, calculate, and get insights easily and we can have a more wide view on the data frame.

Duration

By subtracting Starting time from Ending time I have calculated the trip duration and saved into the column “duration”. There were some negative durations and I decided to drop the rows with negative values

```
## Adding the new column
data <- data %>%
  mutate(duration = ended_at - started_at)

## Looking for negative values
negative_durations <- filter(data, duration < 0)
negative_durations
```

```
##             ride_id rideable_type          started_at            ended_at
## 1  7C1E92200AEFF70E   docked_bike 2020-04-27 17:20:30 2020-04-27 17:20:18
## 2  671BB1F73F4CD303   docked_bike 2020-04-20 16:51:18 2020-04-20 16:51:05
## 3  502B972C6B1FCAE6   docked_bike 2020-04-12 19:09:54 2020-04-12 19:09:40
## 4  68E70FDF06F0A439   docked_bike 2020-04-29 17:37:11 2020-04-29 17:37:07
## 5  6EB323BCC83A9D1D   docked_bike 2020-04-05 15:46:12 2020-04-05 15:46:11
## 6  90105A0FA1F2B0F3   docked_bike 2020-04-13 14:06:59 2020-04-13 14:06:45
## 7  BDFF2212459A9858   docked_bike 2020-04-27 18:51:42 2020-04-27 18:51:14
## 8  BFF9D20C42D3B693   docked_bike 2020-04-19 14:10:16 2020-04-19 14:10:03
## 9  15FE83B5CC494A1C   docked_bike 2020-04-19 18:52:35 2020-04-19 18:52:32
## 10 F6FDF112F975A216   docked_bike 2020-04-28 15:35:24 2020-04-28 15:35:11
## 11 00ED4786F962B827   docked_bike 2020-04-28 06:55:20 2020-04-28 06:54:33
## 12 3C19503CC3A81CCE   docked_bike 2020-04-29 16:54:01 2020-04-29 16:51:05
## 13 83A8FE824A191902   docked_bike 2020-04-14 07:52:00 2020-04-14 07:51:48
## 14 B99C5A0F67262E04   docked_bike 2020-04-28 15:45:05 2020-04-28 15:45:00
## 15 ADC76DBE67BAAD66   docked_bike 2020-04-29 17:41:39 2020-04-29 17:41:33
## 16 DF401D495C8822E1   docked_bike 2020-04-24 16:19:11 2020-04-24 16:18:53
## 17 12608C026A6E01C4   docked_bike 2020-04-29 17:50:14 2020-04-29 17:50:02
## 18 41E80A7BDB6409C3   docked_bike 2020-04-24 17:59:12 2020-04-24 17:59:00
## 19 6FC11E831B21B28D   docked_bike 2020-04-27 18:49:58 2020-04-27 18:49:25
## 20 D6092C5E242F6D47   docked_bike 2020-04-15 19:12:09 2020-04-15 19:12:04
## 21 72E721ADC38364D2   docked_bike 2020-04-14 07:53:26 2020-04-14 07:53:22
## 22 6F90CC047E2C55E2   docked_bike 2020-04-13 18:12:58 2020-04-13 18:12:44
## 23 05B000481136CAF5   docked_bike 2020-04-24 17:49:36 2020-04-24 17:49:19
## 24 11195C3052EE09B2   docked_bike 2020-04-11 16:39:48 2020-04-11 16:39:33
## 25 E28382CB814CAD8C   docked_bike 2020-04-28 06:53:39 2020-04-28 06:53:32
## 26 C6F50A326A5F883E   docked_bike 2020-04-19 18:51:46 2020-04-19 18:51:10
## 27 4C4F851B25D81BBC   docked_bike 2020-04-29 13:34:01 2020-04-29 13:33:56
## 28 B32C8EF3CE9CC515   docked_bike 2020-04-01 14:22:11 2020-04-01 14:22:00
## 29 A9C677AB30627686   docked_bike 2020-04-15 19:15:44 2020-04-15 19:15:15
## 30 ED9B2819C03EAA96   docked_bike 2020-04-15 19:09:04 2020-04-15 19:08:52
## 31 AD5373DC1F4D6B59   docked_bike 2020-04-01 14:18:09 2020-04-01 14:17:26
## 32 60E7DF06C9297609   docked_bike 2020-04-14 07:54:02 2020-04-14 07:53:49
## 33 BD4E690138CDB544   docked_bike 2020-04-13 09:18:43 2020-04-13 09:18:28
## 34 F6F91F2D50F2B535   docked_bike 2020-04-27 18:49:03 2020-04-27 18:47:52
## 35 81F6F85E1A4A35EC   docked_bike 2020-04-28 15:48:01 2020-04-28 15:47:59
## 36 B4261465811E2A5B   docked_bike 2020-04-28 14:43:33 2020-04-28 14:43:09
## 37 1D114F8E9C600BB8   docked_bike 2020-04-24 16:22:11 2020-04-24 16:21:57
## 38 278E6C16DD4BD347   docked_bike 2020-04-28 14:33:53 2020-04-28 14:33:40
## 39 01DCBDE0B77F5D1F   docked_bike 2020-04-20 16:51:17 2020-04-20 16:51:04
## 40 65F936B734227D02   docked_bike 2020-04-19 13:59:17 2020-04-19 13:59:04
## 41 F2C87277147935E2   docked_bike 2020-04-24 16:28:24 2020-04-24 16:28:15
## 42 D5B59617C88CB993   docked_bike 2020-04-11 15:52:16 2020-04-11 15:52:12
## 43 BA9FA2547D002402   docked_bike 2020-04-12 21:08:00 2020-04-12 21:07:59
## 44 A6681B96BA16F372   docked_bike 2020-04-27 17:30:15 2020-04-27 17:29:54
## 45 F9DC488A3AF6DDB1   docked_bike 2020-04-27 18:44:45 2020-04-27 18:44:14
## 46 6A858185960B73D1   docked_bike 2020-04-19 14:10:39 2020-04-19 14:10:34
## 47 1056D9D5CB3053F4   docked_bike 2020-04-11 15:59:01 2020-04-11 15:58:41
## 48 555C041720DC0A05   docked_bike 2020-04-25 13:55:28 2020-04-25 13:55:20
## 49 C244E78C41553525   docked_bike 2020-04-29 17:51:49 2020-04-29 17:51:18
## 50 DA2362CDEDA0A371   docked_bike 2020-04-18 12:23:21 2020-04-18 12:23:17
## 51 8D63CE7BD65E2650   docked_bike 2020-04-07 07:53:54 2020-04-07 07:53:46
##                         start_station_name start_station_id
```

```
## 1              St. Clair St & Erie St              211
## 2           Lake Shore Dr & Belmont Ave            334
## 3              Clark St & Schiller St             301
## 4              Dearborn St & Adams St              37
## 5           Francisco Ave & Foster Ave            471
## 6             Racine Ave & Belmont Ave            226
## 7         Sheffield Ave & Wellington Ave          115
## 8           Desplaines St & Randolph St            96
## 9             Clark St & Armitage Ave              94
## 10             St. Clair St & Erie St             211
## 11            Orleans St & Hubbard St             636
## 12          Sheffield Ave & Webster Ave           327
## 13           Indiana Ave & Roosevelt Rd           255
## 14             Michigan Ave & Lake St              52
## 15        Michigan Ave & Washington St             43
## 16          Michigan Ave & Jackson Blvd           284
## 17            Honore St & Division St              17
## 18 HUBBARD ST BIKE CHECKING (LBS-WH-TEST)         671
## 19               Broadway & Ridge Ave             461
## 20           Sheridan Rd & Columbia Ave           660
## 21            Wabash Ave & Roosevelt Rd            59
## 22             Clark St & Schiller St             301
## 23 HUBBARD ST BIKE CHECKING (LBS-WH-TEST)         671
## 24            Damen Ave & Sunnyside Ave           316
## 25        Sheffield Ave & Wrightwood Ave          302
## 26           Kedzie Ave & Milwaukee Ave           260
## 27            Cherry Ave & Blackhawk St           666
## 28             Loomis St & Jackson Blvd           146
## 29            Halsted St & Clybourn Ave           331
## 30             Clark St & Lincoln Ave             141
## 31                  Clark St & Elm St             176
## 32               Broadway & Argyle St             295
## 33         Ravenswood Ave & Lawrence Ave          344
## 34            Columbus Dr & Randolph St           195
## 35            Sheffield Ave & Willow St            93
## 36            Clark St & Wellington Ave           156
## 37            Clinton St & Jackson Blvd           638
## 38        Southport Ave & Wrightwood Ave          190
## 39            Wacker Dr & Washington St            18
## 40             Manor Ave & Leland Ave             477
## 41                 Ellis Ave & 58th St            328
## 42          Western Ave & Winnebago Ave           116
## 43             State St & Harrison St               5
## 44            Southport Ave & Roscoe St           229
## 45         Milwaukee Ave & Wabansia Ave           158
## 46               Halsted St & 18th St             202
## 47        Ravenswood Ave & Irving Park Rd         244
## 48           Sheridan Rd & Lawrence Ave           323
## 49                  Wells St & Elm St             182
## 50             McClurg Ct & Illinois St            26
## 51            Wood St & Milwaukee Ave              61
##                               end_station_name end_station_id start_lat
## 1             Clinton St & Washington Blvd             91   41.8944
## 2                      Buckingham Fountain              2   41.9408
```

```
## 3                        Wells St & Concord Ln          289   41.9080
## 4                     California Ave & 23rd Pl          442   41.8794
## 5                     Damen Ave & Clybourn Ave          163   41.9756
## 6                      Broadway & Waveland Ave          304   41.9397
## 7                 Michigan Ave & Washington St           43   41.9363
## 8                     Halsted St & Dickens Ave          225   41.8846
## 9          Orleans St & Merchandise Mart Plaza          100   41.9183
## 10                    Fairbanks Ct & Grand Ave           24   41.8944
## 11                         Wells St & Huron St           53   41.8900
## 12                   Clark St & Wellington Ave          156   41.9215
## 13                     Clark St & Ida B Wells Dr         50   41.8679
## 14                 Racine Ave & Wrightwood Ave          343   41.8860
## 15                Michigan Ave & Washington St           43   41.8840
## 16                       Wabash Ave & Adams St           39   41.8779
## 17                                 Eckhart Park          86   41.9031
## 18      HUBBARD ST BIKE CHECKING (LBS-WH-TEST)          671   41.8900
## 19                     Clark St & Bryn Mawr Ave         460   41.9840
## 20                    Glenwood Ave & Morse Ave          447   42.0046
## 21             Cityfront Plaza Dr & Pioneer Ct          196   41.8672
## 22                         Federal St & Polk St          41   41.9080
## 23      HUBBARD ST BIKE CHECKING (LBS-WH-TEST)          671   41.8900
## 24                   Damen Ave & Wellington Ave         162   41.9633
## 25                  Fairbanks St & Superior St          635   41.9287
## 26                Humboldt Blvd & Armitage Ave          507   41.9296
## 27                  Cherry Ave & Blackhawk St           666   41.9072
## 28                        Green St & Madison St         198   41.8779
## 29                         Wells St & Huron St           53   41.9097
## 30                            Clark St & Elm St         176   41.9157
## 31                   Sedgwick St & Schiller St          236   41.9030
## 32                     Damen Ave & Melrose Ave          228   41.9738
## 33                     Broadway & Cornelia Ave          303   41.9691
## 34                      Clinton St & Madison St          77   41.8847
## 35 Burling St (Halsted) & Diversey Pkwy (Temp)          332   41.9137
## 36                         Clark St & Grace St          165   41.9365
## 37                    Columbus Dr & Randolph St         195   41.8781
## 38                    Leavitt St & Armitage Ave         309   41.9288
## 39                    Columbus Dr & Randolph St         195   41.8831
## 40                      Manor Ave & Leland Ave          477   41.9659
## 41                      Lake Park Ave & 53rd St         419   41.7887
## 42                      Halsted St & Willow St          224   41.9155
## 43                    Fairbanks Ct & Grand Ave           24   41.8741
## 44              Southport Ave & Wellington Ave          153   41.9437
## 45         Damen Ave & Thomas St (Augusta Blvd)         183   41.9126
## 46                       Clinton St & 18th St           170   41.8575
## 47               Ravenswood Ave & Berteau Ave           314   41.9547
## 48                Sheridan Rd & Irving Park Rd          240   41.9695
## 49                            Wells St & Elm St         182   41.9032
## 50                      Streeter Dr & Grand Ave          35   41.8904
## 51                     Wood St & Milwaukee Ave           61   41.9077
##     start_lng end_lat  end_lng member_casual  duration
## 1    -87.6227 41.8834 -87.6412        member  -12 secs
## 2    -87.6392 41.8765 -87.6205        casual  -13 secs
## 3    -87.6315 41.9121 -87.6347        member  -14 secs
## 4    -87.6298 41.8491 -87.6951        member   -4 secs
```

```
## 5   -87.7014 41.9319 -87.6779        member   -1 secs
## 6   -87.6589 41.9491 -87.6486        casual  -14 secs
## 7   -87.6527 41.8840 -87.6247        member  -28 secs
## 8   -87.6446 41.9199 -87.6488        casual  -13 secs
## 9   -87.6363 41.8882 -87.6364        member   -3 secs
## 10  -87.6227 41.8918 -87.6206        member  -13 secs
## 11  -87.6366 41.8947 -87.6344        member  -47 secs
## 12  -87.6538 41.9365 -87.6475        member -176 secs
## 13  -87.6230 41.8759 -87.6306        member  -12 secs
## 14  -87.6241 41.9289 -87.6590        member   -5 secs
## 15  -87.6247 41.8840 -87.6247        casual   -6 secs
## 16  -87.6241 41.8795 -87.6257        member  -18 secs
## 17  -87.6739 41.8964 -87.6610        member  -12 secs
## 18  -87.6807 41.8900 -87.6807        casual  -12 secs
## 19  -87.6603 41.9836 -87.6692        casual  -33 secs
## 20  -87.6614 42.0080 -87.6655        casual   -5 secs
## 21  -87.6260 41.8906 -87.6221        member   -4 secs
## 22  -87.6315 41.8721 -87.6295        casual  -14 secs
## 23  -87.6807 41.8900 -87.6807        casual  -17 secs
## 24  -87.6793 41.9359 -87.6784        member  -15 secs
## 25  -87.6538 41.8957 -87.6201        member   -7 secs
## 26  -87.7079 41.9175 -87.7018        member  -36 secs
## 27  -87.6556 41.9072 -87.6556        casual   -5 secs
## 28  -87.6620 41.8819 -87.6488        member  -11 secs
## 29  -87.6481 41.8947 -87.6344        member  -29 secs
## 30  -87.6346 41.9030 -87.6313        casual  -12 secs
## 31  -87.6313 41.9076 -87.6386        member  -43 secs
## 32  -87.6597 41.9406 -87.6785        member  -13 secs
## 33  -87.6742 41.9455 -87.6464        casual  -15 secs
## 34  -87.6195 41.8822 -87.6411        member  -71 secs
## 35  -87.6529 41.9331 -87.6478        member   -2 secs
## 36  -87.6475 41.9508 -87.6592        casual  -24 secs
## 37  -87.6398 41.8847 -87.6195        member  -14 secs
## 38  -87.6639 41.9178 -87.6824        member  -13 secs
## 39  -87.6373 41.8847 -87.6195        member  -13 secs
## 40  -87.7008 41.9659 -87.7008        member  -13 secs
## 41  -87.6013 41.7995 -87.5864        member   -9 secs
## 42  -87.6870 41.9139 -87.6488        member   -4 secs
## 43  -87.6277 41.8918 -87.6206        casual   -1 secs
## 44  -87.6640 41.9357 -87.6636        casual  -21 secs
## 45  -87.6814 41.9013 -87.6774        member  -31 secs
## 46  -87.6463 41.8580 -87.6408        member   -5 secs
## 47  -87.6739 41.9579 -87.6736        casual  -20 secs
## 48  -87.6547 41.9542 -87.6544        casual   -8 secs
## 49  -87.6343 41.9032 -87.6343        member  -31 secs
## 50  -87.6175 41.8923 -87.6120        member   -4 secs
## 51  -87.6726 41.9077 -87.6726        casual   -8 secs
```

```
## Filtering and saving into a new dataframe
filtered_data <- filter(data, duration > 0)
```

Weekdays

Categorizing the dates by weekdays and day of the week and labeling them to save in a “day_of_the_week” column

```
## Numerically identifying the weekdays starting with Monday -> 1
filtered_data <- filtered_data %>%
  mutate(day_of_the_week = wday(filtered_data$started_at, week_start = 1))

## With the defined values we can filter and label the weekdays and weekends
filtered_data <- filtered_data %>%
  mutate(weekday = ifelse(day_of_the_week >= 6, "weekend", "weekday"))
```

Months

Columns with year and Month added as “year_month”

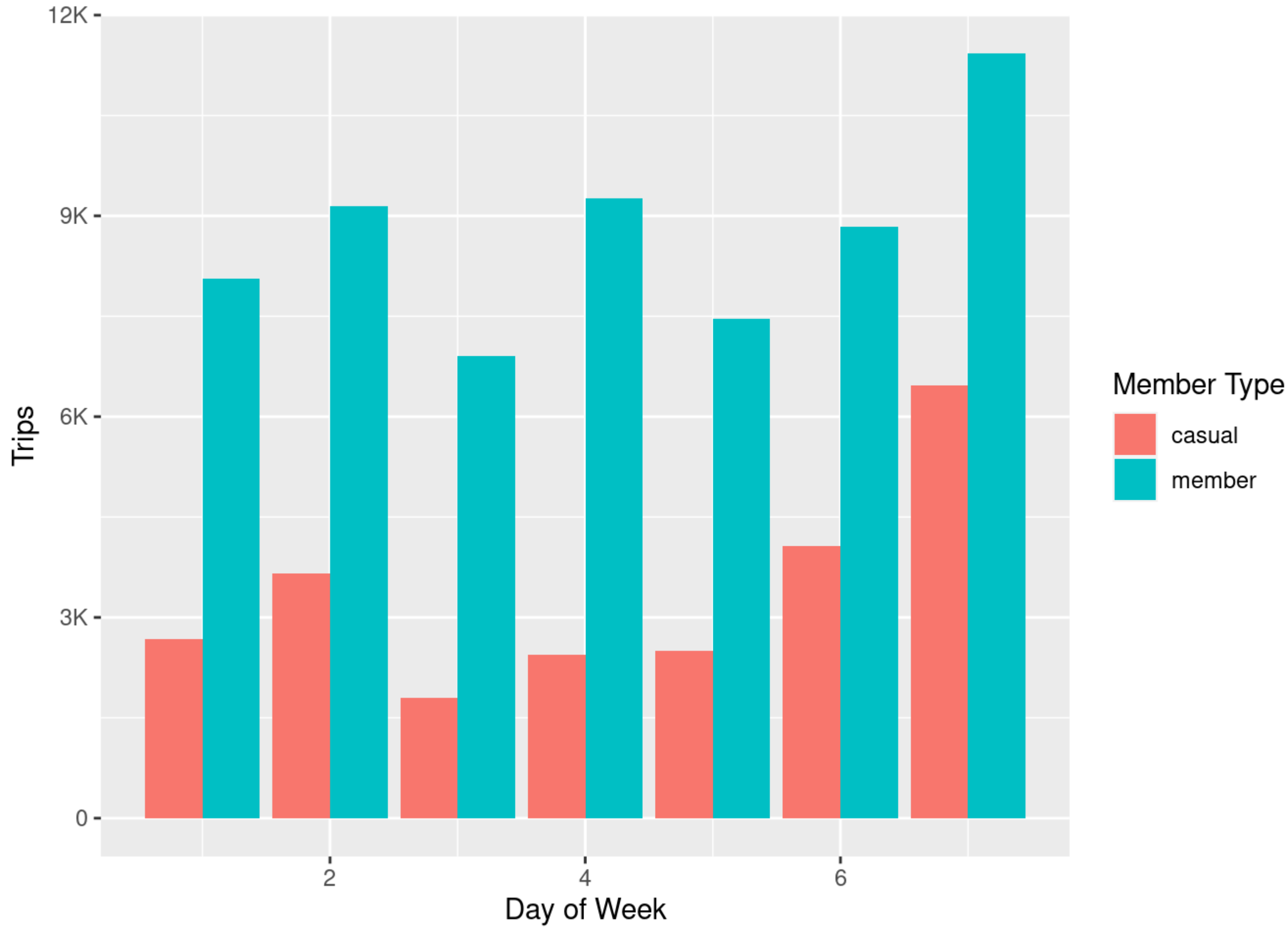
```
filtered_data <- filtered_data %>%
  mutate(year_month = format(as.POSIXct(filtered_data$started_at), format = "%Y%m"))
```

Plots

With plots, we can visually examine the relations of the data.

Member Type vs the Day of the week

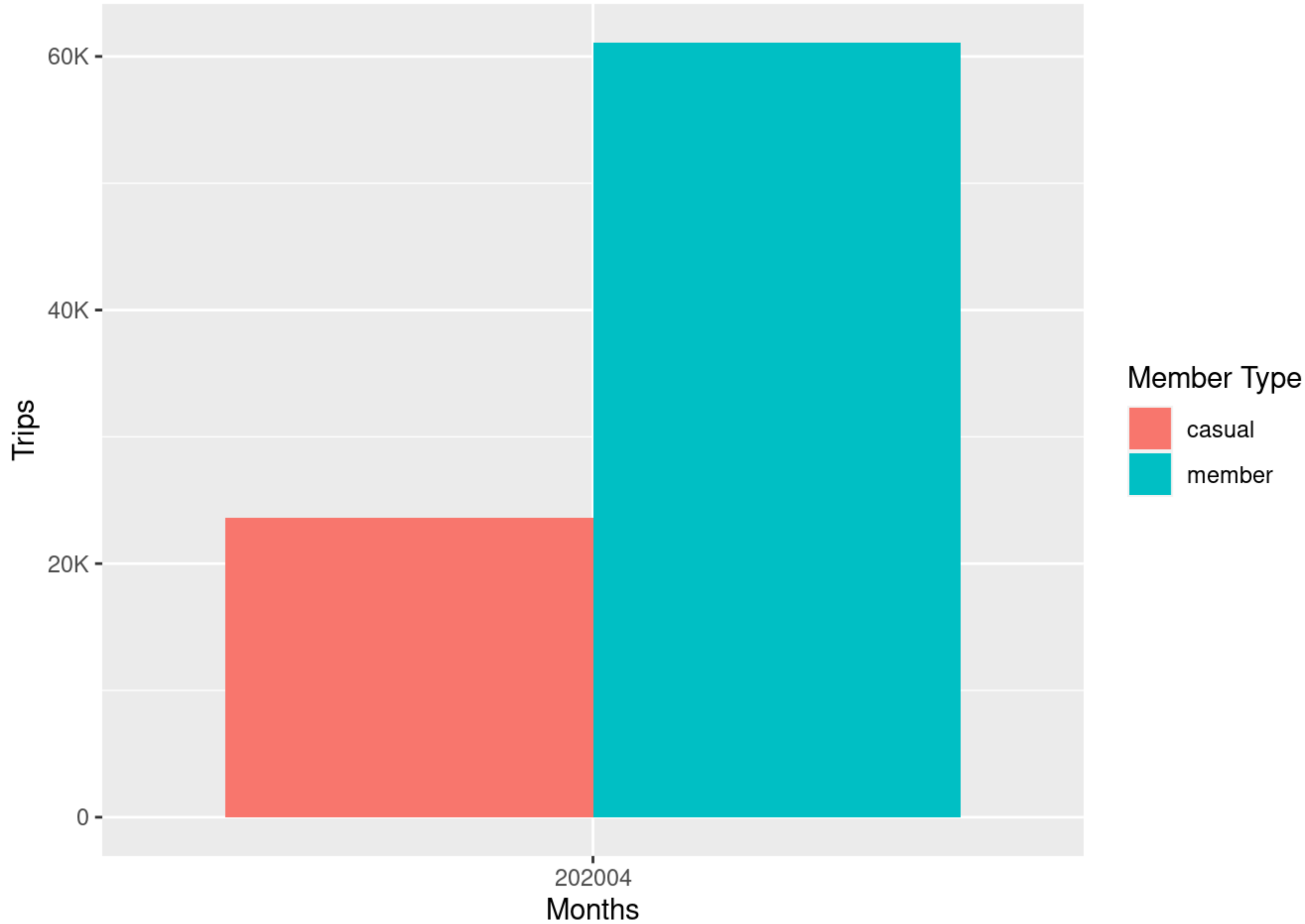
```
ggplot(data=filtered_data, aes(fill=member_casual, x=day_of_the_week)) +
  geom_bar(position="dodge", stat="count") +
  scale_y_continuous(labels = scales::label_number_si()) +
  theme(axis.text.x = element_text()) +
  labs(y = "Trips", x = "Day of Week", fill = "Member Type")
```



Number of trips are increasing for both member types till Saturday and sudden fall in Sunday than Saturday but not more than Friday. Overall, weekends have the highest number of trips. For the weekdays, there is a recognizable gap between two member types and the trips. But in the weekend casual members have increased almost twice the number than weekdays. How ever we can see that permenent members have the highest number of trips throughout the week.

Member Type vs the Month distribution

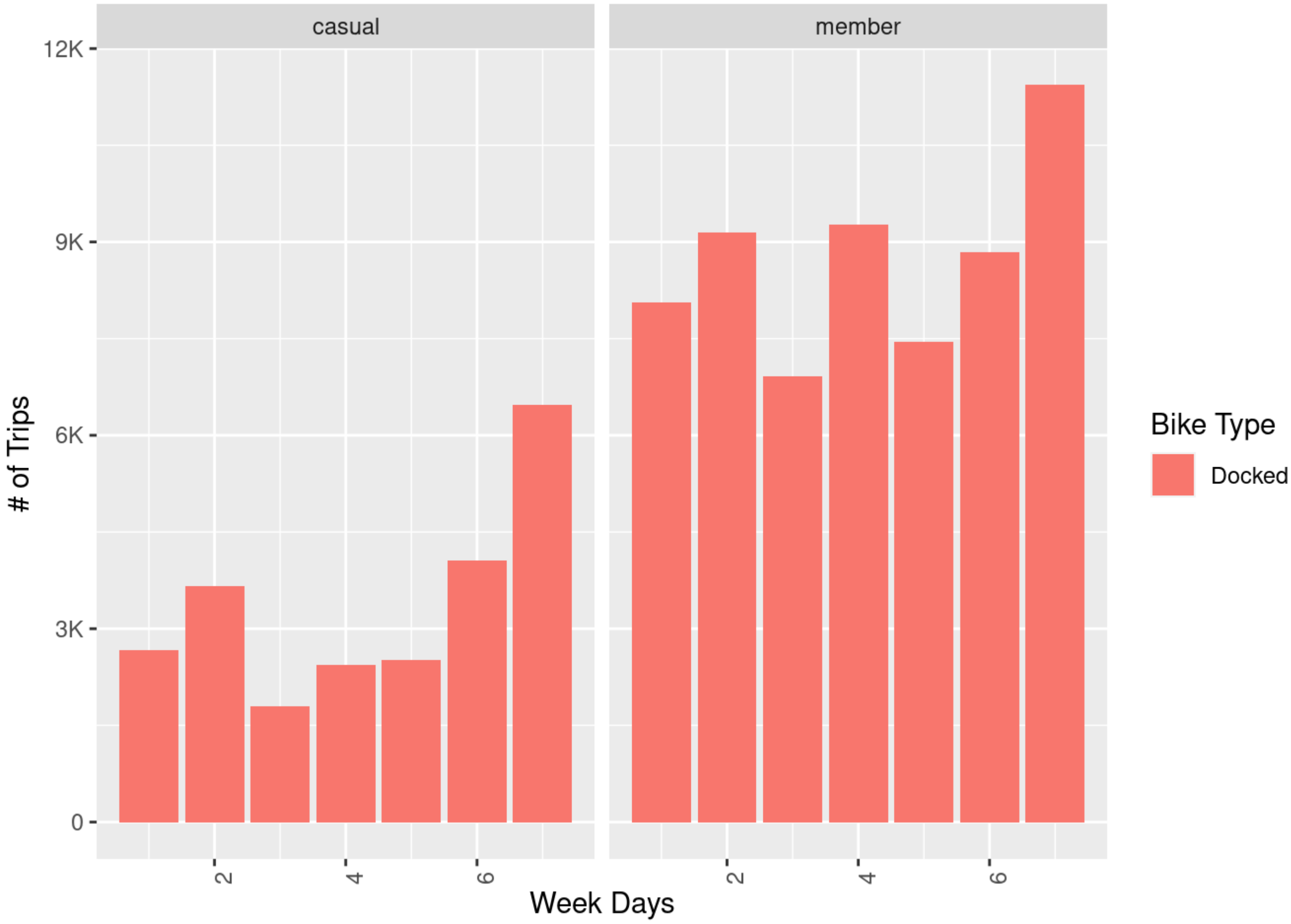
```
ggplot(data=filtered_data, aes(fill=member_casual, x=year_month)) +
  geom_bar(position="dodge", stat="count") +
  scale_y_continuous(labels = scales::label_number_si()) +
  theme(axis.text.x = element_text()) +
  labs(y = "Trips", x = "Months", fill = "Member Type")
```



We can see that number of trips are increasing in a seasonal pattern. Casual member trips are higher in some months.

Member Type vs Bike Type distribution

```
ggplot(data=filtered_data, aes(fill=rideable_type, x=day_of_the_week)) +
  geom_bar(position="stack", stat="count") +
  scale_y_continuous(labels = scales::label_number_si()) +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(y = "# of Trips", x = "Week Days") +
  scale_fill_discrete(name="Bike Type", labels = c("Docked","Electric")) +
  facet_wrap(~member_casual)
```



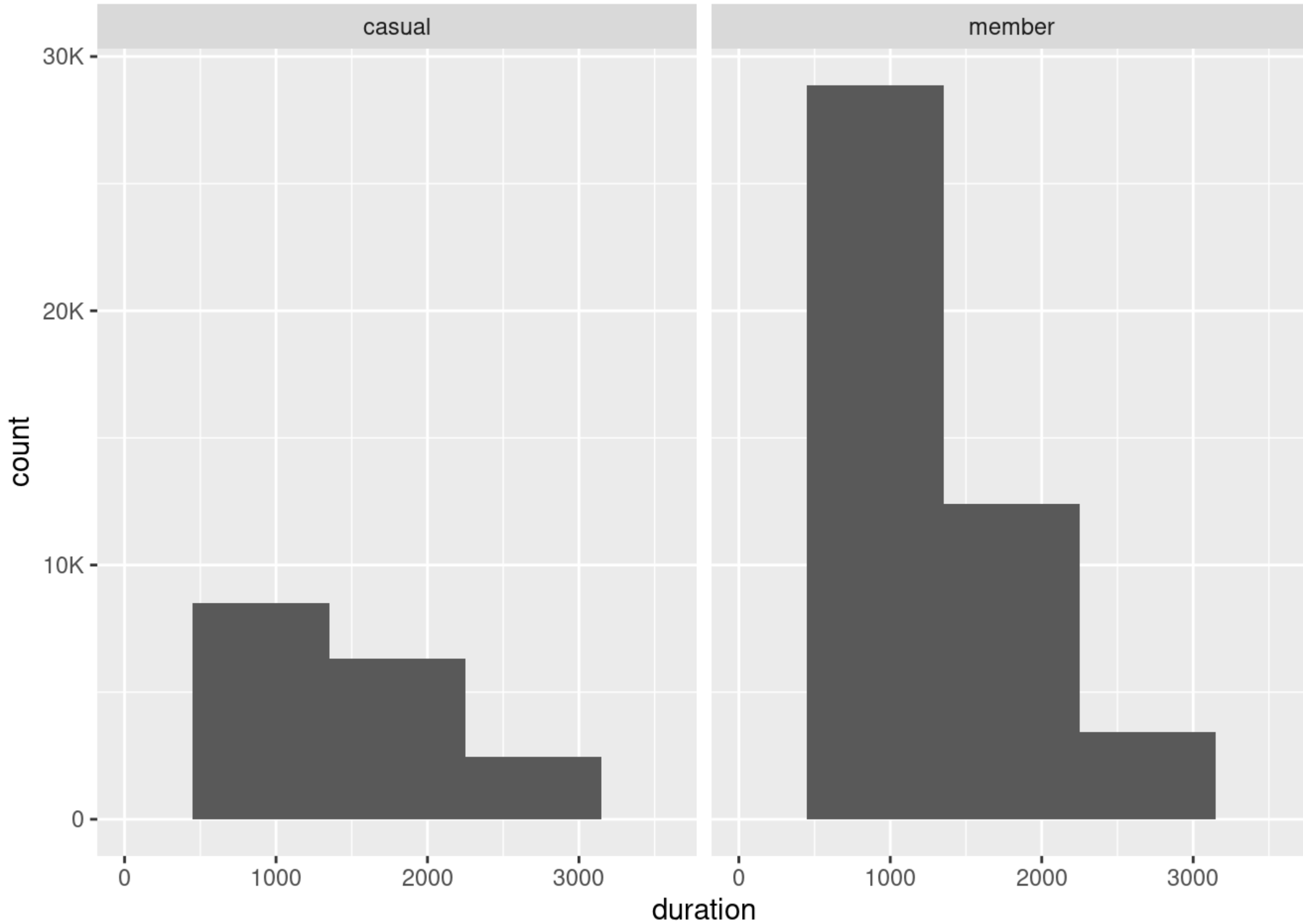
Docked bikes have the upper hand on popularity over electric bikes. As per the above plot usage is increasing towards the weekends.

Durations with Member types

```
ggplot(data=filtered_data, aes(duration)) +
  geom_histogram(binwidth = 900) +
  xlim(0, 3600) +
  scale_y_continuous(labels = scales::label_number_si()) +
  facet_wrap(~member_casual)
```

```
## Warning: Removed 4477 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```



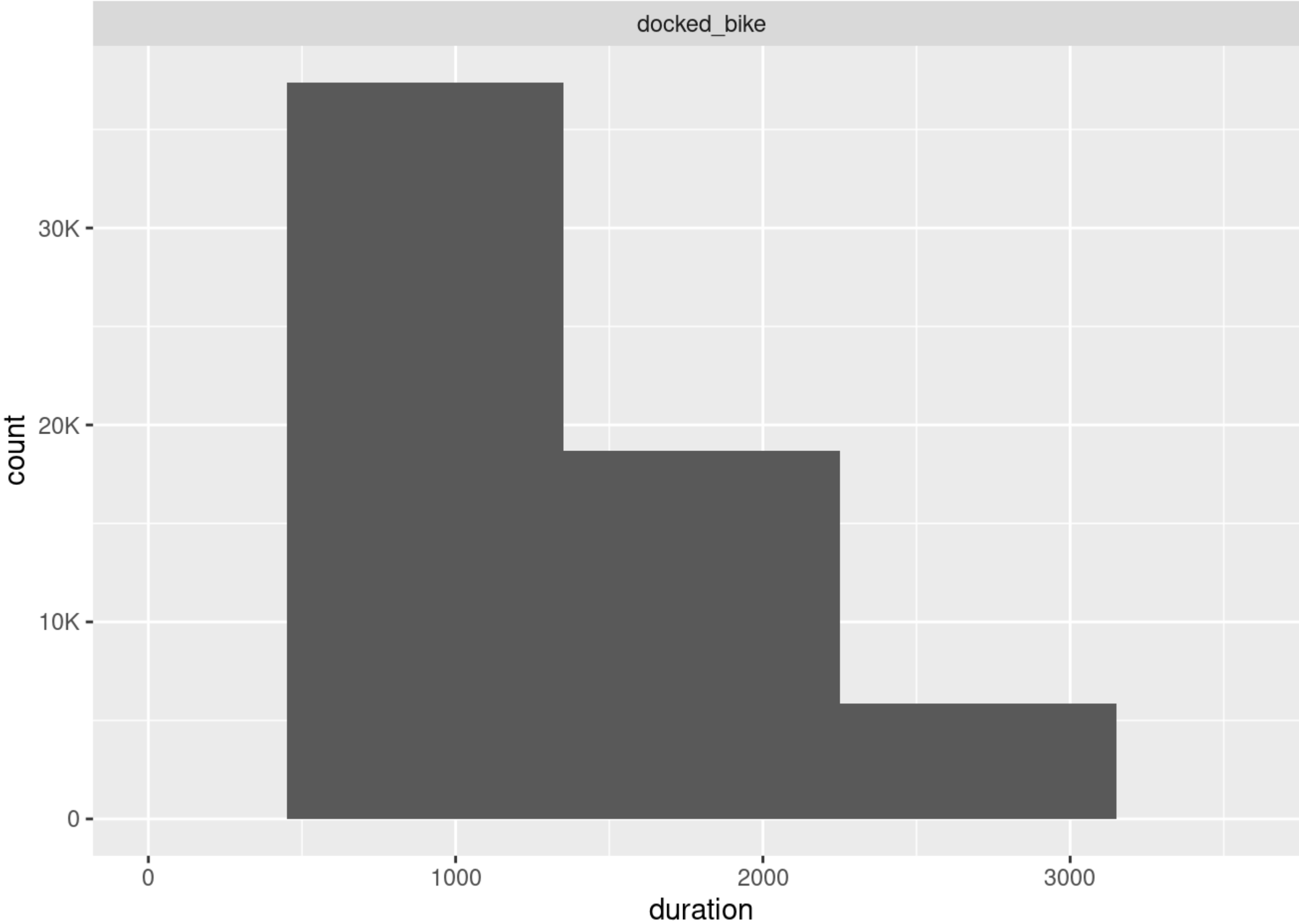
Patterns are same in the two plots. But we can see alarming high value in permanent members plot around 1000 secs.

Durations with Bike types

```
ggplot(data=filtered_data, aes(duration)) +
  geom_histogram(binwidth = 900) +
  xlim(0, 3600) +
  scale_y_continuous(labels = scales::label_number_si()) +
  facet_wrap(~rideable_type)
```

```
## Warning: Removed 4477 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



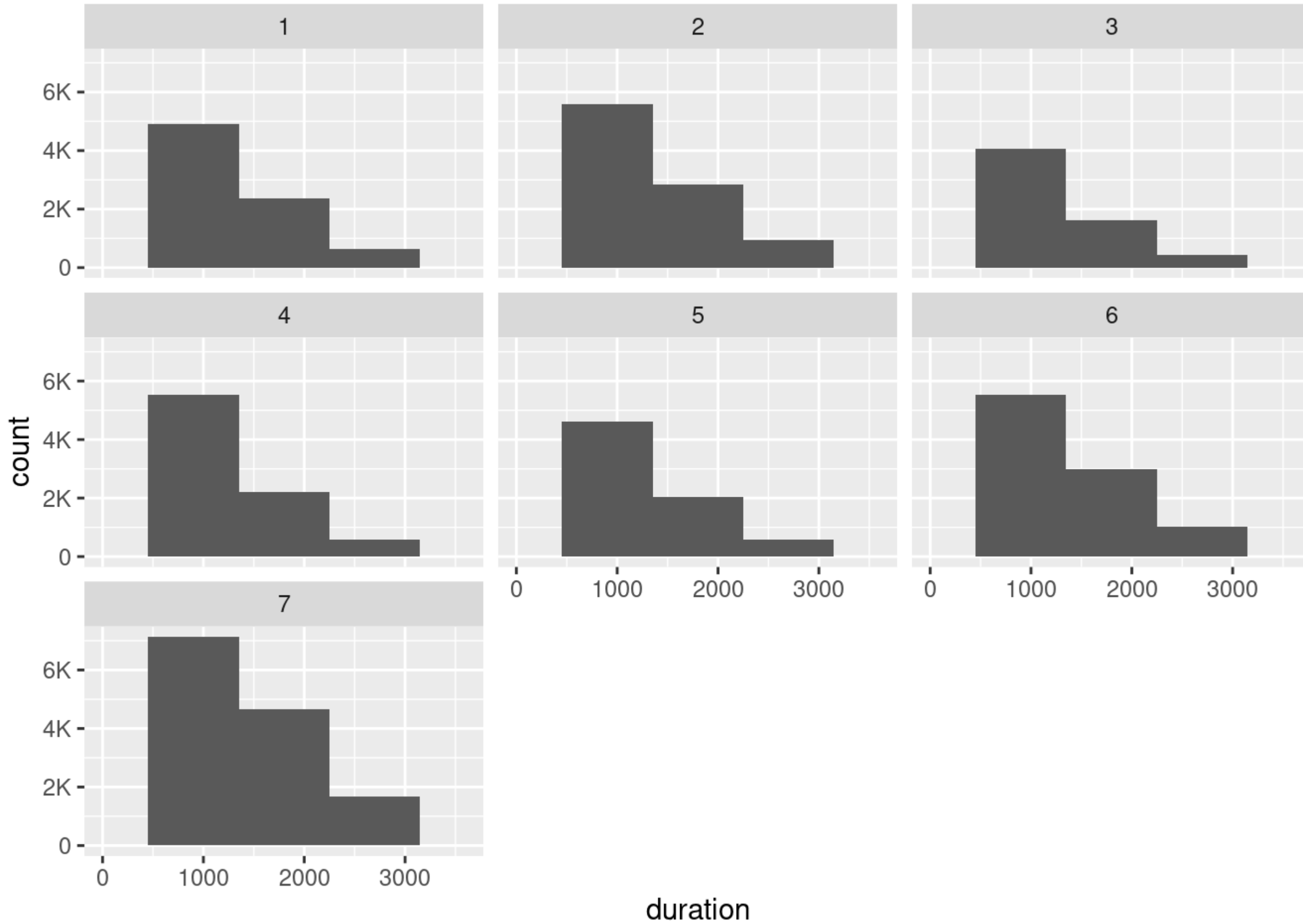
Both bike types have same patterns in the plots.

Duration with Day of the week

```
ggplot(data=filtered_data, aes(duration)) +
  geom_histogram(binwidth = 900) +
  xlim(0, 3600) +
  scale_y_continuous(labels = scales::label_number_si()) +
  facet_wrap(~day_of_the_week)
```

```
## Warning: Removed 4477 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 14 rows containing missing values (geom_bar).
```



All the patterns have same decreasing pattern with duration. We can see a increase in 3000 secs duration in the weekend plots which could be a valuable point to identify the audience.

Top 10 Stations

Top stations among Casual members

```
start_stations <- filtered_data %>%
  filter(start_station_name != "") %>%
  filter(member_casual == "casual") %>%
  group_by(start_station_name) %>%
  tally(sort = TRUE) %>%
  ungroup() %>%
  arrange(desc(n))
head(start_stations, 10)
```

```
## # A tibble: 10 × 2
##    start_station_name              n
##    <chr>                       <int>
##  1 Clark St & Elm St             246
##  2 Wabash Ave & Grand Ave        206
##  3 Desplaines St & Kinzie St     204
##  4 Wells St & Huron St           204
##  5 Clark St & Lincoln Ave        198
##  6 Wells St & Elm St             195
##  7 Sheffield Ave & Waveland Ave  191
##  8 Dearborn St & Erie St         185
##  9 Ashland Ave & Division St     182
## 10 Clark St & Armitage Ave       175
```

Top 10 stations among Permenant members

```
start_stations <- filtered_data %>%
  filter(start_station_name != "") %>%
  filter(member_casual == "member") %>%
  group_by(start_station_name) %>%
  tally(sort = TRUE) %>%
  ungroup() %>%
  arrange(desc(n))
head(start_stations, 10)
```

```
## # A tibble: 10 × 2
##    start_station_name              n
##    <chr>                       <int>
##  1 Clark St & Elm St             602
##  2 St. Clair St & Erie St        591
##  3 Dearborn St & Erie St         545
##  4 Desplaines St & Kinzie St     516
##  5 Clark St & Armitage Ave       450
##  6 Broadway & Barry Ave          444
##  7 Stockton Dr & Wrightwood Ave  420
##  8 Wabash Ave & Grand Ave        408
##  9 Clark St & Schiller St        406
## 10 Larrabee St & Webster Ave     404
```

Only Clark St & Elm St, Clark St & Lincoln Ave, Larrabee St & Webster Ave stations are common in top 10 stations of the both member types

Top 10 stations among Docked bike users

```
start_stations <- filtered_data %>%
  filter(start_station_name != "") %>%
  filter(rideable_type == "docked_bike") %>%
  group_by(start_station_name) %>%
  tally(sort = TRUE) %>%
  ungroup() %>%
  arrange(desc(n))
head(start_stations, 10)
```

```
## # A tibble: 10 × 2
##    start_station_name              n
##    <chr>                       <int>
##  1 Clark St & Elm St             848
##  2 Dearborn St & Erie St         730
##  3 Desplaines St & Kinzie St     720
##  4 St. Clair St & Erie St        684
##  5 Clark St & Armitage Ave       625
##  6 Wabash Ave & Grand Ave        614
##  7 Broadway & Barry Ave          605
##  8 Stockton Dr & Wrightwood Ave  584
##  9 Larrabee St & Webster Ave     576
## 10 Clark St & Schiller St        574
```

Top 10 stations among Electric bike users

```
start_stations <- filtered_data %>%
  filter(start_station_name != "") %>%
  filter(rideable_type == "electric") %>%
  group_by(start_station_name) %>%
  tally(sort = TRUE) %>%
  ungroup() %>%
  arrange(desc(n))
head(start_stations, 10)
```

```
## # A tibble: 0 × 2
## # … with 2 variables: start_station_name <chr>, n <int>
```

Clark St & Elm St, Broadway & Barry Ave, Dearborn St & Erie St, Larrabee St & Webster Ave, Clark St & Armitage Ave, and Wells St & Concord Ln stations are common in top 10 stations by the bike types.

With the above plots and lists we can identify the differences between two groups of members and their specialties. With this data we can target the marketing campaign based on stations, months, weekdays, and bike types.

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.