# Capstone Project: Google Data Analytics certificate

Case Study 1: How does Bike-Share Navigate Speedy Success?

In this project, I will be analyzing a public dataset for a fictional company named Cyclistic provided by the course. I will use program R for data analysis and Tableau for visualizations.

The following data analysis follow the ask, prepare, process, analyze, share and act steps.

**About the company and Role**

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. Cyclistic's main marketing strategy has been to build general awareness and appealing to broad consumer segments by focusing on the flexibility of its pricing plans. Cyclistic uses three different pricing points such as single-ride passes, full-day passes and annual memberships. Consumers that mainly use single-ride passes and/or full-day passes are referred as casual riders and consumers who purchase annual memberships are referred as Cyclistic members.

A financial analysis was conducted and it was determined that annual members were more profitable than casual riders. The director of the marketing team and manager, Lily Moreno, believes in maximizing the number of annual members by converting casual riders into Cyclistic members rather than advertising new customers to take the membership.

As a junior data analyst for Moreno's team, I am tasked to analyze bike-share data to find trends and patterns for casual riders to better understand how they use the company's bike-share rides to better help Cyclistic to aim at converting casual riders into becoming annual members.

## Ask

**Business Task**: Coming up with marketing strategies to convert casual riders into annual Cyclistic members.

These are the following business questions that will guide the future marketing program:

1. How do annual members and casual riders use Cyclistic bikes differently?
2. Why would casual riders buy Cyclistic annual memberships?
3. How can Cyclistic use digital media to influence casual riders to become members?

As a Junior Data Analyst, I was assigned to answer the first question where my focus will be on comparing the annual members and casual riders.

**Key Stakeholders**: • Lily Moreno – Director of the marketing team as well as my manager • Cyclistic executive team

## Prepare

To conduct a thorough analysis, I have obtained the Cyclistic historical data trips from January 2022 to December 2022. This data is available by Motivate International Inc. under this license.

After downloading the datasets, I have organized them by month and year in a folder named "Cyclistic_rider_data_2022". I then uploaded each month's data into BigQuery under the dataset with the name "Clyclistic".

After examination of the tables, each table contains the same 13 columns and the data is appropriate and consistent throughout. These columns are:

1. Ride_id
2. Rideable_type
3. Started_at
4. Ended_at
5. Start_station_name
6. Start_station_id
7. End_station_name
8. End_station_id
9. Start_lat
10. Start_lng
11. End_lat
12. End_lng
13. Member_casual

**Process**

I will use RStudio, an integrated development environment designed for the programming language R, to clean, analyze and aggregate data collected from Motivate International Inc from January 2022 to December 2022 which will be stored in a folder with the name Cyclistic Data 2022.

To start, I will create a new script or query which will serve as a workspace for executing the necessary operations on the Cyclistic data. Then, I will load each data file to RStudio so that I am able to work the data.

```
df1 <- read.csv("Desktop/Cyclistic Data 2022/df1.csv")
df2 <- read.csv("Desktop/Cyclistic Data 2022/df2.csv")
df3 <- read.csv("Desktop/Cyclistic Data 2022/df3.csv")
df4 <- read.csv("Desktop/Cyclistic Data 2022/df4.csv")
df5 <- read.csv("Desktop/Cyclistic Data 2022/df5.csv")
df6 <- read.csv("Desktop/Cyclistic Data 2022/df6.csv")
df7 <- read.csv("Desktop/Cyclistic Data 2022/df7.csv")
df8 <- read.csv("Desktop/Cyclistic Data 2022/df8.csv")
df9 <- read.csv("Desktop/Cyclistic Data 2022/df9.csv")
df10 <- read.csv("Desktop/Cyclistic Data 2022/df10.csv")
df11 <- read.csv("Desktop/Cyclistic Data 2022/df11.csv")
df12 <- read.csv("Desktop/Cyclistic Data 2022/df12.csv")
```

Next, I will load all the packages I will use to perform the data cleaning and manipulation.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(dplyr)
library(lubridate)
```

Now that I have all files loaded as well as all packages necessary loaded, I will merge all the files into one big table for easy data manipulation by using the rbind( ) function. This new table will be under the name of bikerides.

```
bikerides <- rbind(df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df11,df12)
```

To know the structure of the newly made data frame, I will use the str( ) function:

```
str(bikerides)
```

```
## 'data.frame':    5667717 obs. of  13 variables:
##  $ ride_id           : chr  "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED4191054
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-01-13 11:59:47" "2022-01-10 08:41:56" "2022-01-25 04:53:40" "2022-
##  $ ended_at          : chr  "2022-01-13 12:02:44" "2022-01-10 08:46:17" "2022-01-25 04:58:01" "2022-
##  $ start_station_name: chr  "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & Fu
##  $ start_station_id  : chr  "525" "525" "TA1306000016" "KA1504000151" ...
##  $ end_station_name  : chr  "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave & Fullerton
##  $ end_station_id    : chr  "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...
##  $ start_lat         : num  42 42 41.9 42 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  42 42 41.9 42 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr  "casual" "casual" "member" "casual" ...
```

We can see that our new data frame contains 5,667,717 rows and 13 columns in total. We can also see the columns listed with a preview of the fields each column has.

After verifying that the new table contains all rows and all columns from the previous files, I will move on to cleaning the data to better represent the data needed to answer our question for the project.

This is the code used to clean the data to be ready for analysis.

```
cyclistic_clean_data <- bikerides %>%
  select("ride_id", "rideable_type", "started_at", "ended_at",
         "start_station_name", "end_station_name", "member_casual") %>%
  na.omit() %>%
  mutate(trip_length = as.numeric(difftime(ended_at, started_at, units = "mins")),
         weekday = format(as.Date(started_at), "%A")) %>%
  select("ride_id", "rideable_type", "started_at", "ended_at", "start_station_name", "end_station_name"
         "weekday", "trip_length", "member_casual") %>%
  filter(trip_length >=1, trip_length <= (24*60))
```

I will break the code down into small steps to understand how data cleaning was performed.

**Step 1: Assigning a variable**

cyclistic_clean_data <- bikerides %>%

The line of code above implies that the output of the bikeridesis being passed as the input to the subsequent operations which are applied to it step by step. The new data frame from these operations will then be assigned to the object cyclistic_clean_data, which results in a new cleaned dataset for our analysis.

**Step 2: Selecting the necessary columns for analysis**

select("ride_id", "rideable_type", "started_at", "ended_at", "start_station_name", "end_station_name", "member_casual") %>%

I used the select_( ) function to choose specific columns from the output of the bikerides being passed as an input to the function.

### Step 3: Removing all the missing values

na.omit() %>%

I used the function na.omit( ) to remove all the rows with mussing values (NA) from the selected columns of the bikerides. This step helps us clean and remove data that will affect the analysis.

### Step 4: Creating new columns needed for analysis

Since our business task is to design marketing strategies aimed at converting casual riders to annual members, particularly by comparing how the casual riders and members differ from one another, I can create columns to extract crucial information that can be found in the current data. These columns can provide information such as trip length and weekday. These columns can provide key information such as how often do casual riders and members use bike services each day of the week and the duration of each bike ride for each casual rider and member.

mutate(trip_length = as.numeric(difftime(ended_at, started_at, units = "mins")), weekday = format(as.Date(started_at), "%A")) %>%

I used the mutate( ) function to create two new columns to show trip_length and weekday which was added to the bikerides data frame.

- Trip_length column will be formatted as numeric data type and the rows it contains will be taken from the difference of the ended_at and started_at columns and the output of it will be in minutes.
- Weekday column is created my formatting the started_at column as a weekday using the format( ) function.

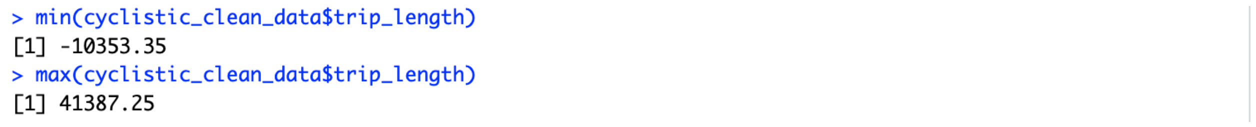
### Step 5: Select the columns with the trip_length and weekday added

select("ride_id", "rideable_type", "started_at", "ended_at", "start_station_name", "end_station_name", "weekday", "trip_length", "member_casual") %>%

**Step 6: Filter the trip_length

Now that I have all the columns needed, I will verify if the trip_length column in ready for analysis. I do this by using the min( ) and max( ) function.

min(cyclistic_clean_data$trip_{l}ength)max(cyclistic_{c}lean_{d}ata$trip_length)



```
> min(cyclistic_clean_data$trip_length)
[1] -10353.35
> max(cyclistic_clean_data$trip_length)
[1] 41387.25
```

The numbers that come up are all in minutes and it appears that the minimum time is negative, which there are no negative time and the maximum is above 24 hours. For this project, trips need to be within 24 hours to make a correct analysis.

To only show trips that are within 24 hours, I will use the filter( ) function and insert it into the cyclistic_clean_data

filter(trip_length >=1, trip_length <= (24*60))

Finally, the resulting clean data frame is stored in a new object named cyclistic_clean_data. This data frame contains the selected and added columns, with missing values removed and filtered values.

To make sure that the data frame is clean and ready, I will recheck and verify if there are still things that are needed to be cleaned. First, I will confirm that there are no NA values in each of the column by using this function:

```r
colSums(is.na(cyclistic_clean_data))
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   end_station_name            weekday        trip_length
##                  0                  0                  0                  0
##      member_casual
##                  0
```

The function shows that there are 0 NA values in each column. Then I will check if there are any duplicate ride_id so that it can be removed.

```r
any(duplicated(cyclistic_clean_data$ride_id))
```

```
## [1] FALSE
```

The result returned FALSE, which means that there are no duplicated in the data frame

Then, I will recheck the minimum and maximum values for the trip_length:

```r
min(cyclistic_clean_data$trip_length)
```

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

```r
max(cyclistic_clean_data$trip_length)
```

```
## [1] 1439.567
```

It looks like the trip_length has been filtered correctly as it is showing minutes within 24 hours. Lastly, we want to check the structure of the new data frame and see if there are data types that do not match the column type.

```r
str(cyclistic_clean_data)
```

```
## 'data.frame':    5541263 obs. of  9 variables:
##  $ ride_id           : chr  "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED4191054
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-01-13 11:59:47" "2022-01-10 08:41:56" "2022-01-25 04:53:40" "2022-
##  $ ended_at          : chr  "2022-01-13 12:02:44" "2022-01-10 08:46:17" "2022-01-25 04:58:01" "2022-
##  $ start_station_name: chr  "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & Fu
##  $ end_station_name  : chr  "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave & Fullerton
##  $ weekday           : chr  "Thursday" "Monday" "Tuesday" "Tuesday" ...
##  $ trip_length       : num  2.95 4.35 4.35 14.93 6.03 ...
##  $ member_casual     : chr  "casual" "casual" "member" "casual" ...
```

From having 5,667,717 rows and 13 columns of raw data, we now have a clean dataset with 5,541,263 rows and 9 columns. We can see that the data types of each column match what we need for our analysis so it is time to start our analyze of our cleaned data.

## Analyze and Share

Now that our data is clean and ready, we will go back to our business question and try to answer it:

*How do annual members and casual riders use Cyclistic bikes differently?*

We can extract key information that will help us answer this question using our data. This is how we can approach it by using the available data that we have cleaned:

### Comparing the number of rides per month

We can analyze the number of rides per month by looking at how they differ between casual riders and members. This can be done by creating a data frame in RStudio extracting the comuns from the cyclistic_clean_data, which are the started_at column (change the datetime to its equivalent month) and the member_casual column. We will be adding a row_count column to count the bike rides each month.

```
#Comparing number of rides per month
month_count = cyclistic_clean_data %>%
  group_by(months = month.name[month(started_at)], member_casual) %>%
  summarize(row_count = n()) %>%
  arrange(match(months,month.name))
```

```
## `summarise()` has grouped output by 'months'. You can override using the
## `.groups` argument.
```

```
#save cvs for tableau
write.csv(month_count, "Desktop/Cyclistic Data 2022/month count.csv", row.names=FALSE)
```

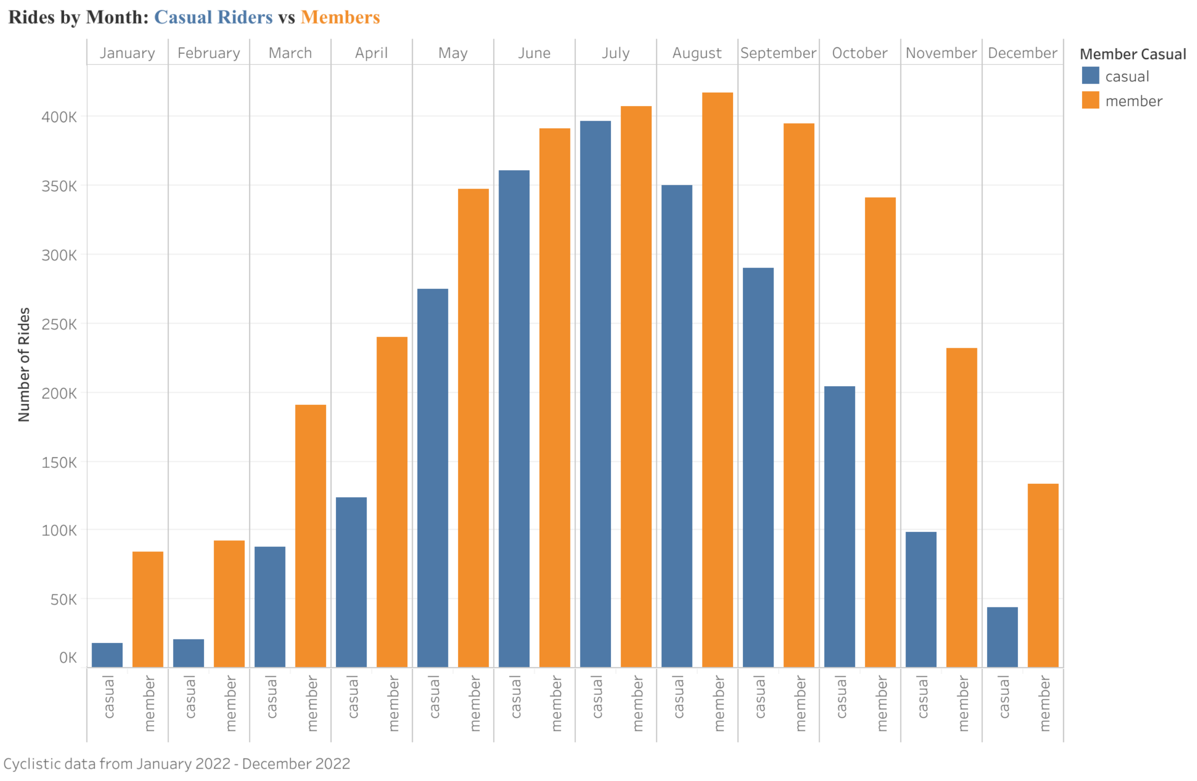
We now created a data frame named month_count. We can use the head( ) function to preview the first 6 rows of the data frame:

```
head(month_count)
```

```
## # A tibble: 6 x 3
## # Groups:   months [3]
##   months   member_casual row_count
##   <chr>    <chr>             <int>
## 1 January  casual            18087
## 2 January  member            83744
## 3 February casual            20917
## 4 February member            92046
## 5 March    casual            88116
## 6 March    member           190566
```

We can see that the data frame is grouped by months by the months column and the member_casual column, this shows the total number of rides per month of the casual riders and members with the row_count column.

To have a better understanding of what these numbers mean, I will use Tableau to visualize the data as followed:



With the Tableau visualization, we can see that the months of June, July and August have the highest number of rides for both casual riders and members. This pattern suggests that during the summer season, where there is typically less rainfall, bike usage tends to increase, attracting both casual riders and members to use bike services more frequently.

**Comparing bike usage on days of the week.**

We can look at how casual riders and members differ in their bike usage on different days of the week by using our clean data. We will extract and group the weekday and member_casual column, create a new column to count the number of rides on a certain weekday, and put in a data frame named weekday_count by using our cyclistic_clean_data data frame.

```
weekday_count = cyclistic_clean_data %>%
  group_by(weekday = weekday, member_casual = member_casual) %>%
  summarize(row_count = n())
```

```
## `summarise()` has grouped output by 'weekday'. You can override using the
## `.groups` argument.
```

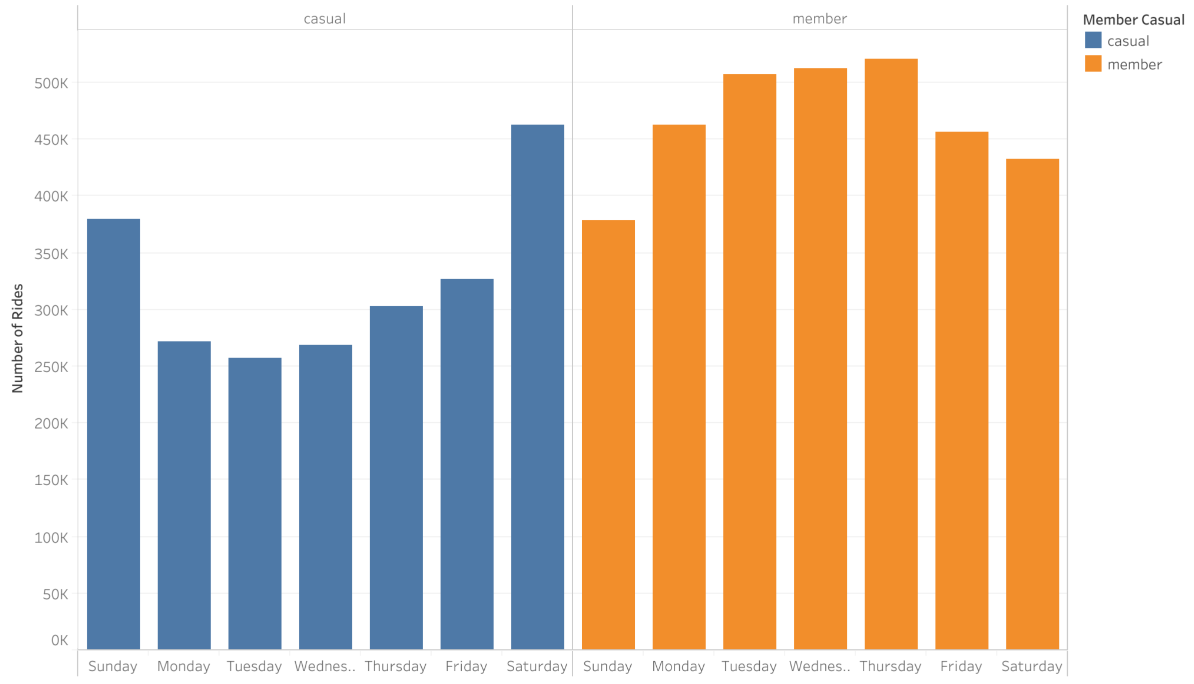
We can view the newly data frame named weekday_count with the View( ) function:

```
head(weekday_count)
```

```
## # A tibble: 6 x 3
## # Groups:   weekday [3]
##   weekday  member_casual row_count
##   <chr>    <chr>             <int>
## 1 Friday   casual           327074
## 2 Friday   member           456820
## 3 Monday   casual           271513
## 4 Monday   member           463174
## 5 Saturday casual           462340
## 6 Saturday member           432747
```

To get a better visualization of the data shown, we will use Tableau to see what the trend is for the casual riders and members:

**Weekly Ride Usage: Casual Riders vs Members**



Cyclistic data from January 2022 - December 2022

By analyzing the graph, we can clearly observe a distinct different in bike usage between casual riders and members.

- **Casual rider's bike usage peaks on weekends:** This could indicate that casual riders are the type of people who likely use the bikes for the purpose of leisure and recreational activities during their free time. This suggests that casual riders use bikes most for personal endeavors such as exploring the city.
- **Members' bike usage shows a peak during weekdays:** This could indicate that members are the type of people who likely use the bikes for transportation and for commuting needs. This suggests that members find bike services to be more convenient and reliable for their transportation for work and weekday commitments.

### Top 5 starting and ending stations

We can identify the top five starting and ending stations for both casual riders and members. This helps us identify popular locations where both groups like to ride. To starts, we will be creating a new data frame and extract the start_station_name, end_station_name, and member_casual columns.

```r
#Start and End stations
top_start_station <- cyclistic_clean_data %>%
  group_by(start_station_name, member_casual) %>%
  summarize(row_count = n()) %>%
  arrange(desc(row_count))
```

```
## `summarise()` has grouped output by 'start_station_name'. You can override
## using the `.groups` argument.
```

```r
head(top_start_station)
```

```
## # A tibble: 6 x 3
## # Groups:   start_station_name [5]
##   start_station_name                  member_casual row_count
##   <chr>                               <chr>             <int>
## 1 ""                                  member           470146
## 2 ""                                  casual          331958
## 3 "Streeter Dr & Grand Ave"           casual           56948
## 4 "DuSable Lake Shore Dr & Monroe St" casual           31241
## 5 "Millennium Park"                   casual           24981
## 6 "Michigan Ave & Oak St"             casual           24834
```

```
top_end_station <- cyclistic_clean_data %>%
  group_by(end_station_name, member_casual) %>%
  summarize(row_count = n()) %>%
  arrange(desc(row_count))
```

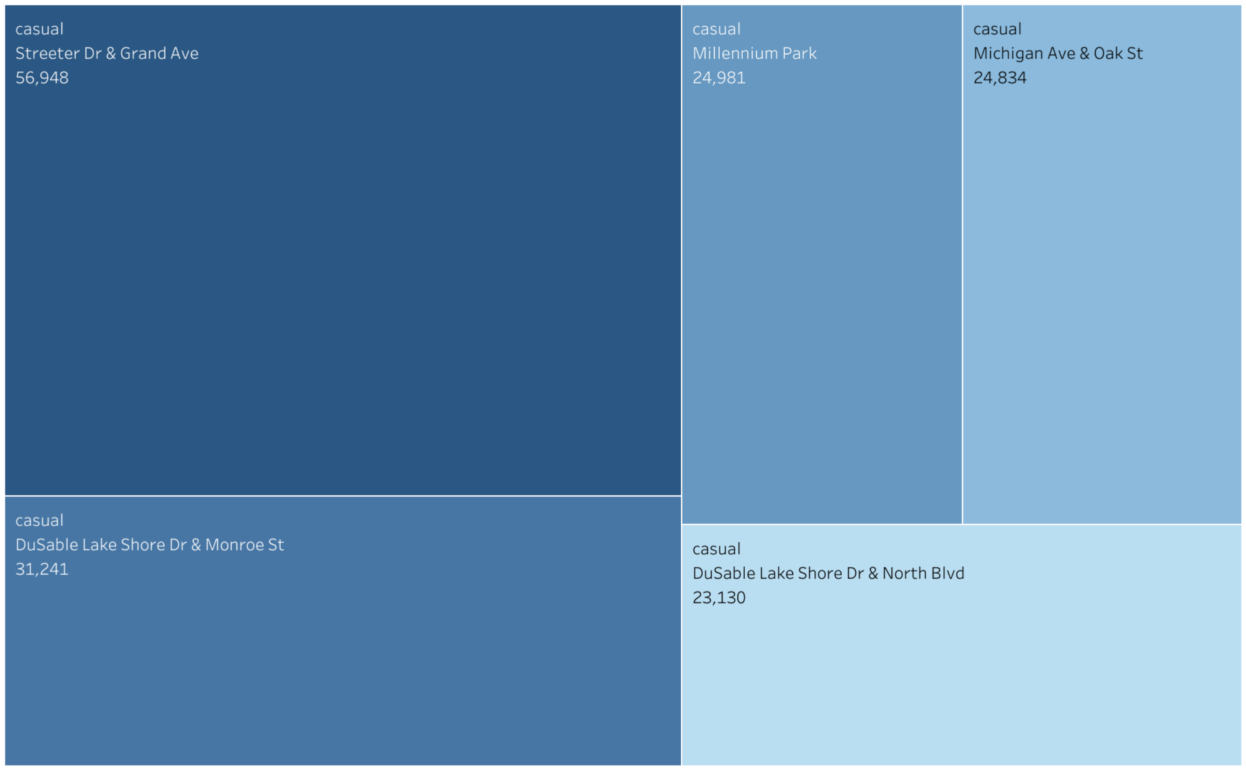
```
## `summarise()` has grouped output by 'end_station_name'. You can override using
## the `.groups` argument.
```

```
head(top_end_station)
```

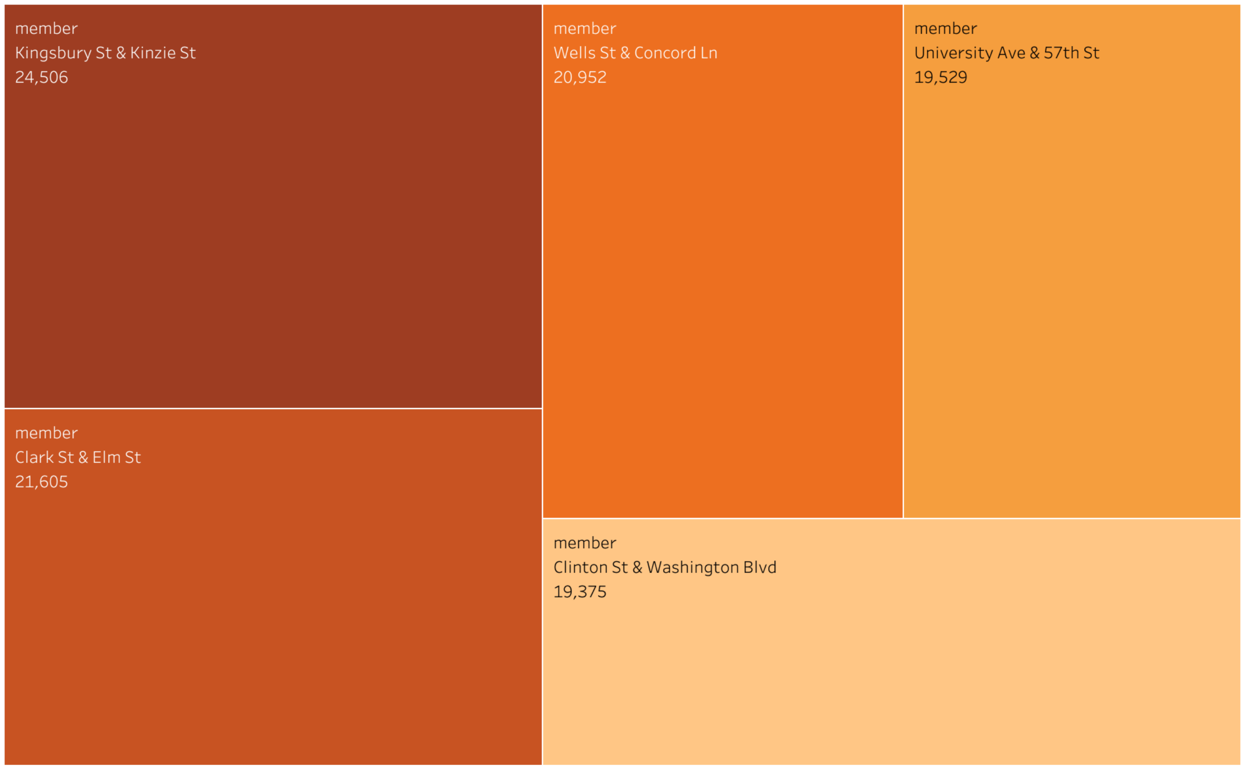
```
## # A tibble: 6 x 3
## # Groups:   end_station_name [5]
##   end_station_name                    member_casual row_count
##   <chr>                               <chr>             <int>
## 1 ""                                  member           461308
## 2 ""                                  casual           383945
## 3 "Streeter Dr & Grand Ave"           casual            58949
## 4 "DuSable Lake Shore Dr & Monroe St" casual            29114
## 5 "Millennium Park"                   casual            26292
## 6 "Michigan Ave & Oak St"             casual            26089
```

The tables show all the locations available in descending order, however, it does not show the top 5 locations for each group. To determine the top 5 locations for each group, we will filter the data in Tableau.

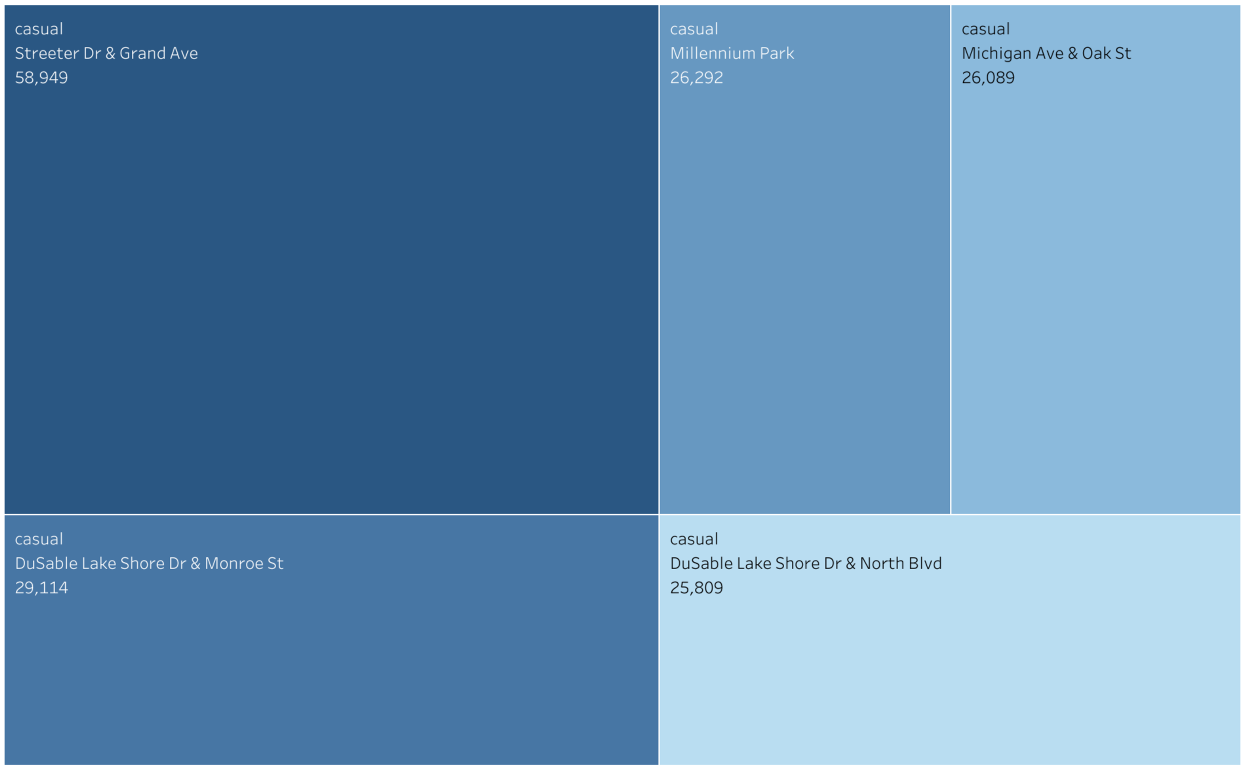
**Top Five Start Stations For Casual Riders**



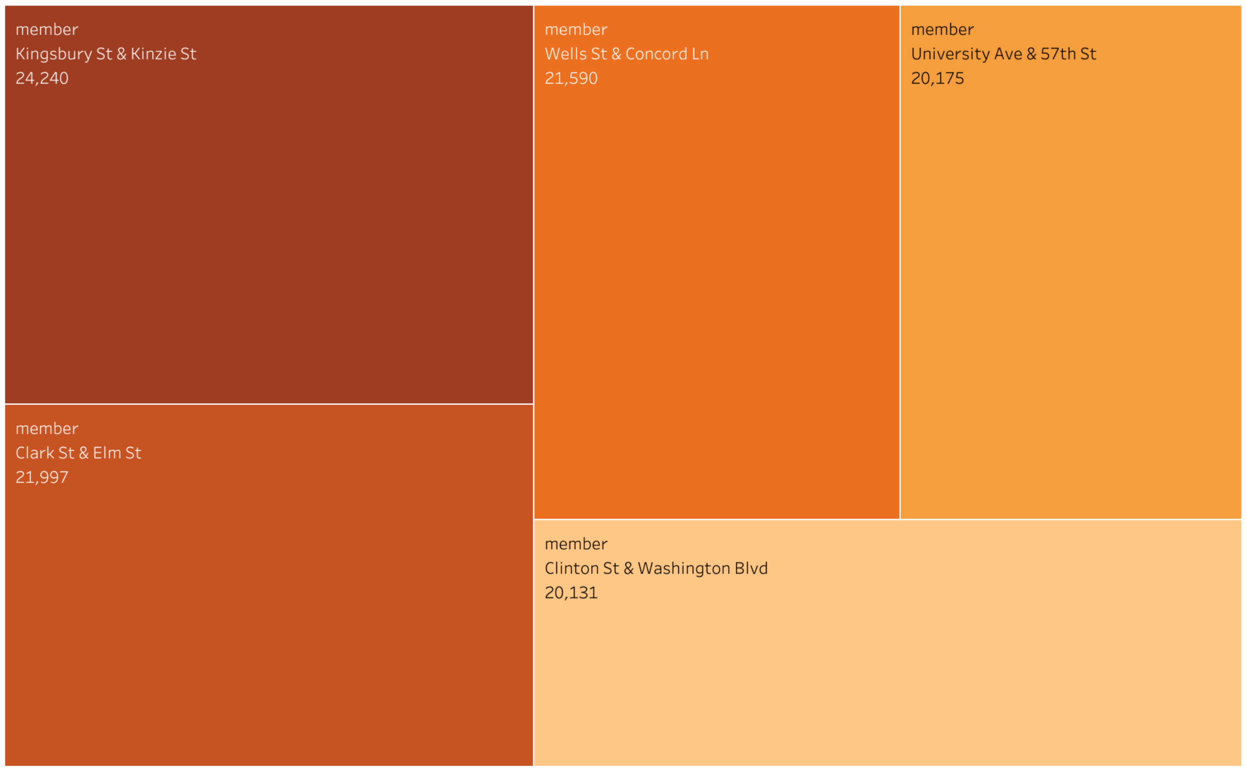
## Top Five Start Stations For Members



## Top Five End Stations for Casual Riders



**Top Five End Stations For Members**



Upon analyzing the heatmaps, it shows that both casual riders and members exhibit distinct top 5 stations. Also, it is interesting to note that the starting stations and ending stations for each group are identical.

This suggests that while casual riders and members have different specific stations, this consistency implies that certain stations serve as a popular and convenient hubs for bike usage across the network.

### Trip Length

By comparing the trip length of casual riders and annual members, it can provide insights into the average duration of the rides for each group and whether there are any differences in ride lengths worth noting.

To conduct this analysis, I will use this code to compare the trip lengths of the two groups:

```r
#Trip Length
ride_length <- cyclistic_clean_data %>%
  group_by(member_casual) %>%
  summarize(mean(trip_length))
```

```r
head(ride_length)
```

```
## # A tibble: 2 x 2
##   member_casual `mean(trip_length)`
##   <chr>                       <dbl>
## 1 casual                       22.3
## 2 member                       12.7
```

Based on this data frame, we can see that casual riders tend to have a longer ride length compared to members with an average trip length of 23 minutes while members average 12 minutes.

This gap in ride lengths between casual riders and members could be pointed to our early analysis when we compared bike usage on different days of the week. For casual riders, this trip length goes in hand with the most activity being during the week which indicates that casual riders use bikes mostly for leisure, activities, exploration or even longer recreational trips which results in longer trip length. Members, on the other hand, the trip length goes hand in hand with more activity being performed during the weekdays which could indicate that members use their bikes for regular commuting for work or for errands as well as shorter transportation needs which tends to have shorter trip length.

### Act

After conducting the analysis of the data, we have learned and gained so much insight into the differences between casual riders and members by looking at their bike usage both during the year and during the week, start and end stations and trip length.

Casual riders use bike services for recreational purposes and leisure, they most likely use bikes to get around town for exploration and for activities while members use bikes to get around town and for commuting. Based on these findings, these are my recommendations to better convert casual riders to members:

1. Seasonal Campaigns/Advertising: Since the most active months for casual riders are the summer months such as June, July and August where casual ridership is at its highest, I would recommend conducting advertising and campaigns to promote the benefits of annual membership by highlighting that a membership provides a more convenient and cost-effective solution during the summer season.
2. Station Membership Promotions: Casual riders mostly use bike rides for leisure and membership promotions centered around popular attractions can encourage casual riders to convert to members. These promotions would be more effective posted on the top 5 start and end stations as these are the places that most casual riders frequent.
3. Weekend Benefits: When looking at activity during the week, casual riders' peak days are the weekend days. Offer benefits on weekends such as coupons for restaurants or shops around each start and end stations as well as exclusive access to certain events by highlighting how convenient and flexible Cyclistic bikes can be for weekend adventures.

This ends my data-driven journey. I hope you enjoyed following along with me and found my analysis insightful. Any feedback is highly appreciated.