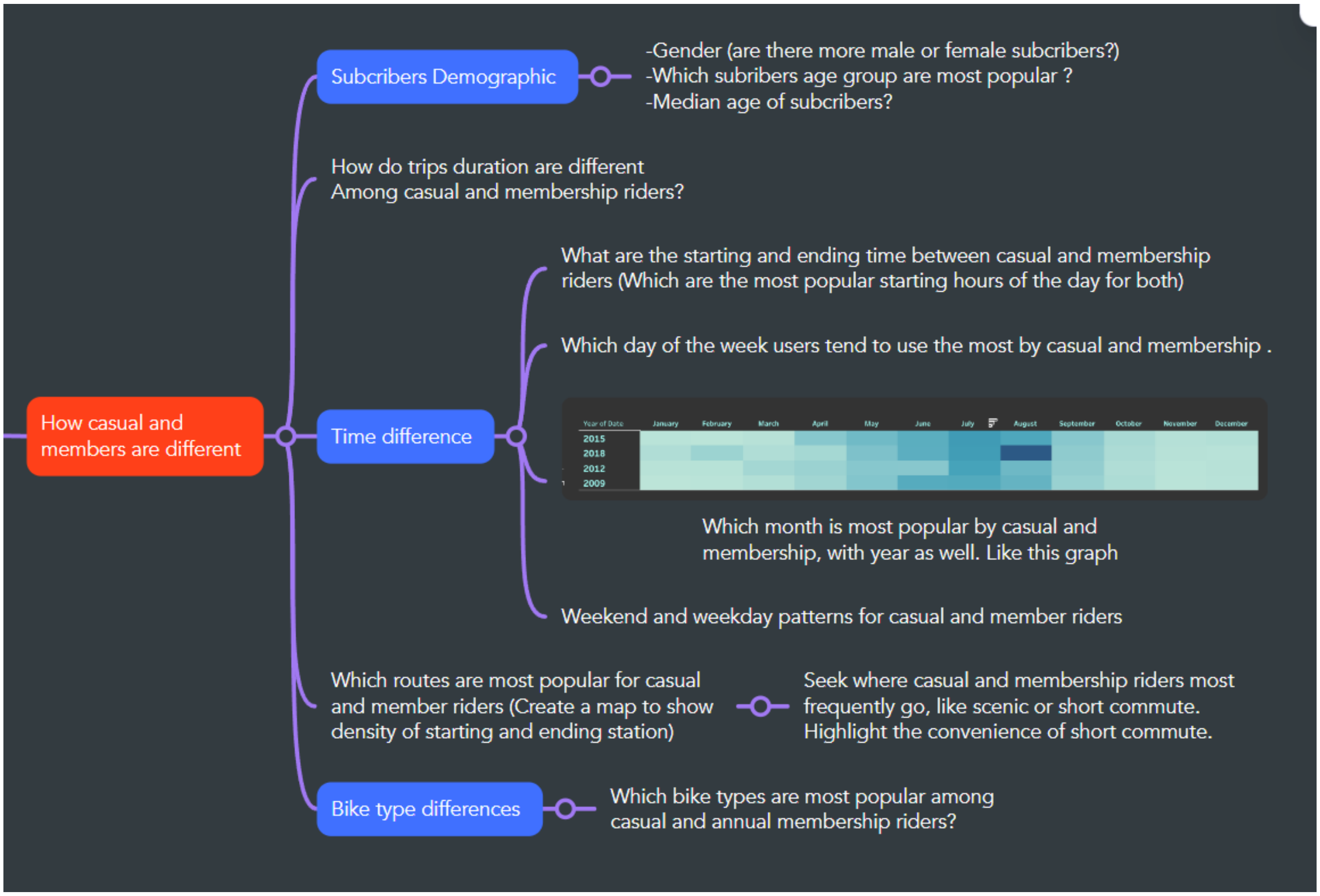
Data Analysis

Objectives

My goal is to identify how do annual members and casual riders use Cyclistic bikes differently? These are my initial insights that may be helpful. All the queries results are stored in Excel_Tableau.Xlsx for Tableau Dashboard.



How casual and members are different

Subcribers Demographic

-Gender (are there more male or female subcribers?)
-Which subribers age group are most popular ?
-Median age of subcribers?

How do trips duration are different
Among casual and membership riders?

Time difference

What are the starting and ending time between casual and membership riders (Which are the most popular starting hours of the day for both)

Which day of the week users tend to use the most by casual and membership .

Which month is most popular by casual and membership, with year as well. Like this graph

Weekend and weekday patterns for casual and member riders

Which routes are most popular for casual and member riders (Create a map to show density of starting and ending station)

Seek where casual and membership riders most frequently go, like scenic or short commute. Highlight the convenience of short commute.

Bike type differences

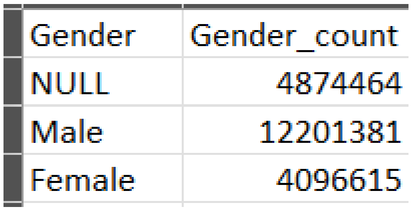
Which bike types are most popular among casual and annual membership riders?

Subscribers Demographic

First, I will Identify what is the dominant gender for Subscribers. Note that for all the data results after querying, I put it to Excel sheet ‘Excel_Tableau.csv’ to visualize in Tableau.

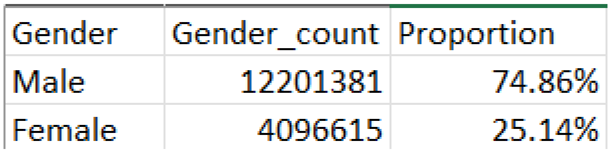
```
/*DATA ANALYSIS */

--What is the dominant gender for the subscribers. Data are in Trips_2013_2019
SELECT Gender,
       COUNT (Gender) AS Gender_count
FROM Trips_2013_2019
GROUP BY Gender
```

We would have



Gender	Gender_count
NULL	4874464
Male	12201381
Female	4096615

But since our focus is to find whether male or female proportion, we can drop the nulls values and get the male and female proportion.



Gender	Gender_count	Proportion
Male	12201381	74.86%
Female	4096615	25.14%

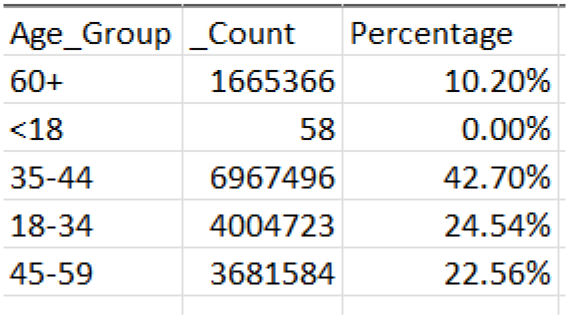
Next, I want to explore the age group and median age of the subscribers.

```
/*  What is the most popular age group? I create an extra age group column
       < 18
       18-35
       35-45
       45-60
       60 +

 */
 --Create a temp table holding Trips_2013_2019
 SELECT  Birth_Year INTO #Age_group
FROM Trips_2013_2019
WHERE Birth_Year is not null

--Add the age group column
ALTER TABLE #Age_group
ADD Age_Group nvarchar(50),
       Age smallint

--Get the Age from the BirthYear Column
UPDATE #Age_group
SET Age = 2024 - Birth_Year

--Create an age group columns based on the Age column
UPDATE #Age_group
SET Age_Group = CASE
    WHEN Age < 18 THEN '<18'
    WHEN Age BETWEEN 18 AND 34 THEN '18-34'
    WHEN Age BETWEEN 35 AND 44 THEN '35-44'
    WHEN Age BETWEEN 45 AND 59 THEN '45-59'
    ELSE '60+'
END;
```

Since there are some users errors when typing the answer, I consider filter out any users > 100 years old.

```
--Identify the Age Group for the subscribers
SELECT Age_Group, COUNT(Age_Group) AS _Count
FROM #Age_group
WHERE AGE < 100
GROUP BY Age_Group
```

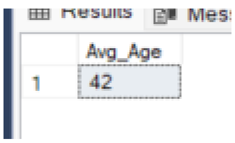
The results



Age_Group	_Count	Percentage
60+	1665366	10.20%
<18	58	0.00%
35-44	6967496	42.70%
18-34	4004723	24.54%
45-59	3681584	22.56%

Now I compute the average age

```
--Compute the median Age for subcribers
SELECT AVG(Age) as Avg_Age
FROM #Age_group
WHERE Age <= 100

--Drop temp table after use
DROP TABLE #Age_group
```



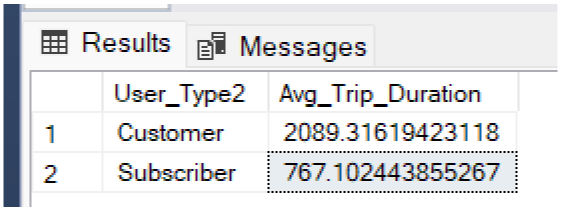
	Avg_Age
1	42

Our Average subscriber age is 42.

Trips duration

```
/* Find the Trips duration for each customer types  */
--Run the #Duration_2020_2023 temp table
SELECT Start_Time, Stop_Time,
    CASE
        WHEN User_Type = 'casual' THEN 'Customer'
        ELSE 'Subscriber'
    END AS User_Type2
INTO #Duration_2020_2023
FROM Trips_2020_2023;

--Compute the average trips duration with Trips_2013_2019 union #Duration_2020_2023
SELECT
    User_Type2,
    AVG(CAST(Trip_Duration AS FLOAT)) AS Avg_Trip_Duration
FROM
(
SELECT DATEDIFF(SECOND, Start_Time, Stop_Time) AS Trip_Duration, User_Type2
FROM #Duration_2020_2023

UNION ALL

SELECT Trip_Duration, User_Type
FROM [dbo].[Trips_2013_2019]
) AS Combined

GROUP BY User_Type2

--Drop temp table after use
DROP TABLE #Duration_2020_2023
```

The results



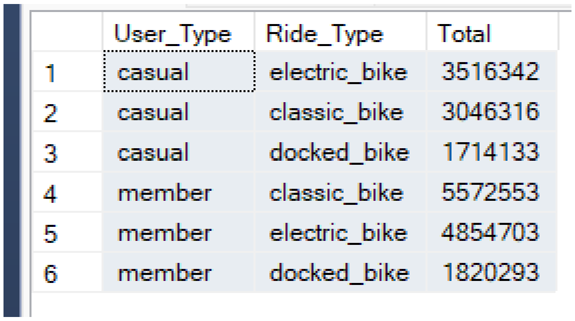
	User_Type2	Avg_Trip_Duration
1	Customer	2089.31619423118
2	Subscriber	767.102443855267

Most popular Bike types

In this part, I aim to identify which type of bike is most popular among casual and member users.

```
--What are the popular Bike Types by User_Type?
SELECT
       User_Type,
       Ride_Type,
       COUNT(Ride_Type) as Total
FROM [dbo].[Trips_2020_2023]

WHERE User_Type is not null

GROUP BY User_Type, Ride_Type
ORDER BY 1, 3 DESC
```

The results



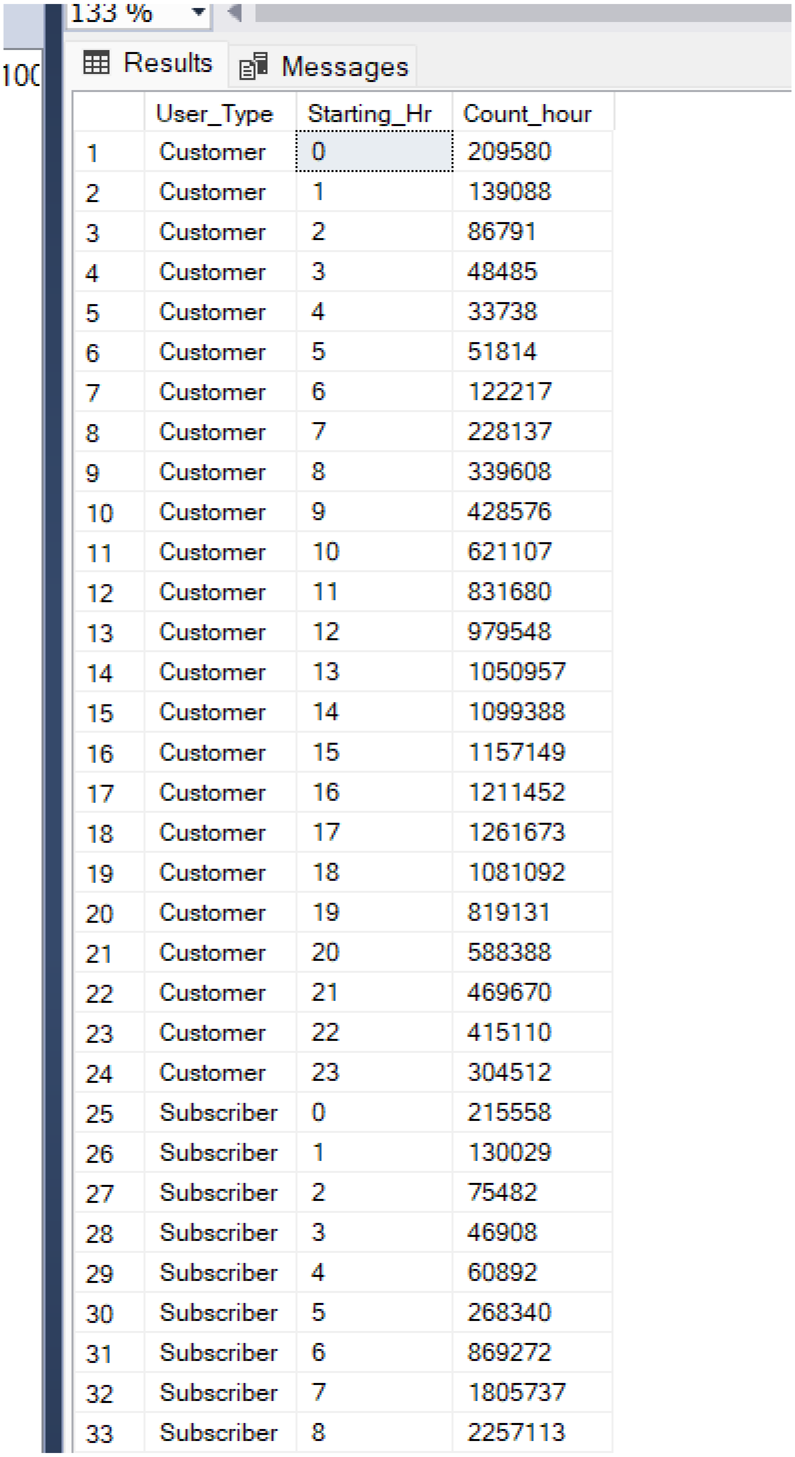
	User_Type	Ride_Type	Total
1	casual	electric_bike	3516342
2	casual	classic_bike	3046316
3	casual	docked_bike	1714133
4	member	classic_bike	5572553
5	member	electric_bike	4854703
6	member	docked_bike	1820293

Time period difference

Firstly, I aim to identify the Starting hours for the user’s type.

```
------------------------------------------------------------------------------------------------
--What are the popular starting time and stop time for user type
--Rerun the temp table
SELECT Start_Time,
    CASE
        WHEN User_Type = 'casual' THEN 'Customer'
        ELSE 'Subscriber'
    END AS User_Type2
INTO #TempTrips_2020_2023
FROM Trips_2020_2023;


--I layered the subqueries and aggregate the hour count by users type
SELECT
       User_Type,
       Starting_Hr,
       COUNT (Starting_Hr) as Count_hour
FROM (
       SELECT
              DATEPART(HOUR,Start_Time) AS Starting_Hr,
               User_Type


       FROM
              (SELECT Start_Time,  User_Type
              FROM [dbo].[Trips_2013_2019]

              UNION ALL

              SELECT*
              FROM #TempTrips_2020_2023)
              AS Combined2
              ) AS Combined3
GROUP BY User_Type, Starting_Hr
ORDER BY 1, 2,3 DESC
```

The results: I would have a list of hour and their counts with user types that I can later use for visualization.

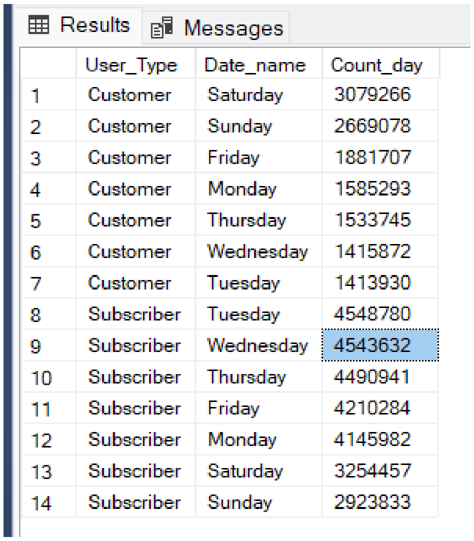


	User_Type	Starting_Hr	Count_hour
1	Customer	0	209580
2	Customer	1	139088
3	Customer	2	86791
4	Customer	3	48485
5	Customer	4	33738
6	Customer	5	51814
7	Customer	6	122217
8	Customer	7	228137
9	Customer	8	339608
10	Customer	9	428576
11	Customer	10	621107
12	Customer	11	831680
13	Customer	12	979548
14	Customer	13	1050957
15	Customer	14	1099388
16	Customer	15	1157149
17	Customer	16	1211452
18	Customer	17	1261673
19	Customer	18	1081092
20	Customer	19	819131
21	Customer	20	588388
22	Customer	21	469670
23	Customer	22	415110
24	Customer	23	304512
25	Subscriber	0	215558
26	Subscriber	1	130029
27	Subscriber	2	75482
28	Subscriber	3	46908
29	Subscriber	4	60892
30	Subscriber	5	268340
31	Subscriber	6	869272
32	Subscriber	7	1805737
33	Subscriber	8	2257113

Next, I want to explore the patterns for day of the week and whether there are differences between weekend and weekday.

```
--Explore weekday and weekend patterns
--Which is the most popular day for the user type

SELECT
       User_Type,
       Date_name,
       COUNT (Date_name) as Count_day
FROM
       (SELECT
              User_Type,
              DATENAME(dw, Start_Time) as Date_name
       FROM
              (SELECT Start_Time,  User_Type
              FROM [dbo].[Trips_2013_2019]

              UNION ALL

              SELECT*
              FROM #TempTrips_2020_2023)
              AS Combined2
              ) AS Combined3
GROUP BY User_Type, Date_name
ORDER BY 1 ,3 DESC
```

The results

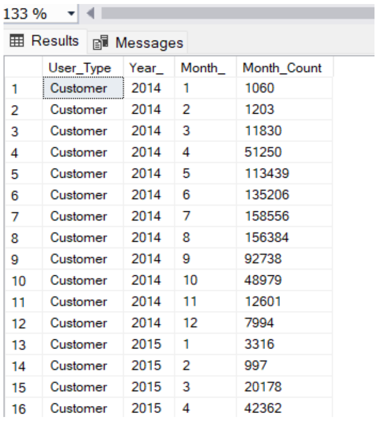


	User_Type	Date_name	Count_day
1	Customer	Saturday	3079266
2	Customer	Sunday	2669078
3	Customer	Friday	1881707
4	Customer	Monday	1585293
5	Customer	Thursday	1533745
6	Customer	Wednesday	1415872
7	Customer	Tuesday	1413930
8	Subscriber	Tuesday	4548780
9	Subscriber	Wednesday	4543632
10	Subscriber	Thursday	4490941
11	Subscriber	Friday	4210284
12	Subscriber	Monday	4145982
13	Subscriber	Saturday	3254457
14	Subscriber	Sunday	2923833

Finally, I want to identify busiest month of the year.

```
--Identify the busiest month of each year
SELECT
       User_Type,
       Year_,
       Month_,
       COUNT(Month_) as Month_Count
FROM
       (SELECT
              User_Type,
              DATEPART(yy, Start_Time) AS Year_,
              DATEPART(m, Start_Time) AS Month_
       FROM
              (SELECT Start_Time,  User_Type
              FROM [dbo].[Trips_2013_2019]

              UNION ALL

              SELECT*
              FROM #TempTrips_2020_2023)
              AS Combined2
              ) AS Combined3
WHERE Year_ NOT IN (2013, 2024) -- As 2013 and 2024 don’t have adequate months
GROUP BY User_Type, Year_, Month_
ORDER BY 1, 2, 3  ASC
```

The results: I would have a list of year and month count for each user types



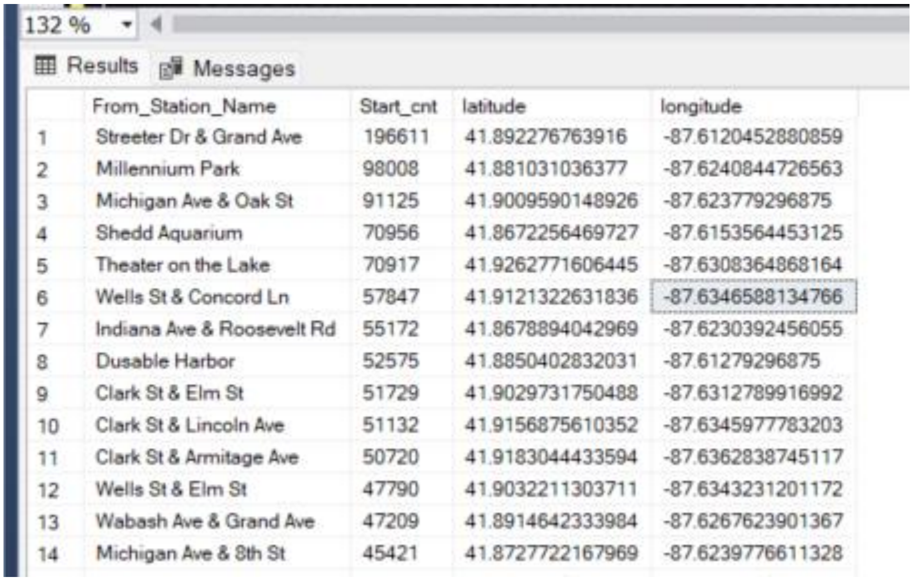
	User_Type	Year_	Month_	Month_Count
1	Customer	2014	1	1060
2	Customer	2014	2	1203
3	Customer	2014	3	11830
4	Customer	2014	4	51250
5	Customer	2014	5	113439
6	Customer	2014	6	135206
7	Customer	2014	7	158556
8	Customer	2014	8	156384
9	Customer	2014	9	92738
10	Customer	2014	10	48979
11	Customer	2014	11	12601
12	Customer	2014	12	7994
13	Customer	2015	1	3316
14	Customer	2015	2	997
15	Customer	2015	3	20178
16	Customer	2015	4	42362

Most popular stations

In the final part, I will explore which starting and ending stations are busiest for casual and annual member customers. Due to incomplete data, I will use from 2020-2023 for the most updated Stations.

```sql
-- Find the most popular starting point for Casual members
SELECT
       From_Station_Name,
       Start_cnt,
       latitude,
       longitude
FROM
(
SELECT
       From_Station_Name,
       COUNT(From_Station_Name) as Start_cnt

FROM [dbo].[Trips_2020_2023]
WHERE User_Type = 'casual'
GROUP BY From_Station_Name
) AS Combined INNER JOIN [dbo].[Station_name] ON
       Combined.From_Station_Name = Station_name.name
ORDER BY Start_cnt DESC
```

The results: we would have a table with starting station name, count of rides, latitude and longitude for casual members. This can be used for visualization later.



	From_Station_Name	Start_cnt	latitude	longitude
1	Streeter Dr & Grand Ave	196611	41.892276763916	-87.6120452880859
2	Millennium Park	98008	41.881031036377	-87.6240844726563
3	Michigan Ave & Oak St	91125	41.9009590148926	-87.623779296875
4	Shedd Aquarium	70956	41.8672256469727	-87.6153564453125
5	Theater on the Lake	70917	41.9262771606445	-87.6308364868164
6	Wells St & Concord Ln	57847	41.9121322631836	-87.6346588134766
7	Indiana Ave & Roosevelt Rd	55172	41.8678894042969	-87.6230392456055
8	Dusable Harbor	52575	41.8850402832031	-87.61279296875
9	Clark St & Elm St	51729	41.9029731750488	-87.6312789916992
10	Clark St & Lincoln Ave	51132	41.9156875610352	-87.6345977783203
11	Clark St & Armitage Ave	50720	41.9183044433594	-87.6362838745117
12	Wells St & Elm St	47790	41.9032211303711	-87.6343231201172
13	Wabash Ave & Grand Ave	47209	41.8914642333984	-87.6267623901367
14	Michigan Ave & 8th St	45421	41.8727722167969	-87.6239776611328

Now repeat to find the ending station for causal members and repeat for Subscriber members

```
-- Find the most popular ending point for Casual members
SELECT
       To_Station_Name,
       To_cnt,
       latitude,
       longitude
FROM
(
SELECT
       To_Station_Name,
       COUNT(To_Station_Name) as To_cnt

FROM [dbo].[Trips_2020_2023]
WHERE User_Type = 'casual'
GROUP BY To_Station_Name
) AS Combined INNER JOIN [dbo].[Station_name] ON
       Combined.To_Station_Name = Station_name.name
ORDER BY To_cnt DESC


-- Find the most popular starting point for Subscriber members
SELECT
       From_Station_Name, Start_cnt, latitude, longitude
FROM
(
SELECT
       From_Station_Name,
       COUNT(From_Station_Name) as Start_cnt

FROM [dbo].[Trips_2020_2023]
WHERE User_Type = 'member'
GROUP BY From_Station_Name
) AS Combined INNER JOIN [dbo].[Station_name] ON
       Combined.From_Station_Name = Station_name.name
ORDER BY Start_cnt DESC


       -- Find the most popular ending point for Subscriber members
SELECT
       To_Station_Name, To_cnt, latitude, longitude
FROM
(
SELECT
       To_Station_Name,
       COUNT(To_Station_Name) as To_cnt

FROM [dbo].[Trips_2020_2023]
WHERE User_Type = 'member'
GROUP BY To_Station_Name
) AS Combined INNER JOIN [dbo].[Station_name] ON
       Combined.To_Station_Name = Station_name.name
ORDER BY To_cnt DESC
```