Business Task

- Analyze how annual members and casual riders use Cyclistic bikes differently. This analysis will help inform marketing strategies aimed at converting casual riders into annual members, which the company believes is key to future growth and profitability.

Data Sources

- The analysis used Cyclistic's historical trip data for the past 12 months, from January to December 2023. The data is public but excludes personally identifiable information due to data-privacy concerns. The datasets contain information such as ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id, end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, and member_casual.

Summary of Cleaning Process

- Added and populated new columns (`ride_length` and `day_of_week`).
- Identified and isolated outliers based on statistical thresholds and missing data.
- Created a cleaned dataset by removing outliers.
- Enhanced the cleaned dataset to handle missing station names by combining coordinates.

Documentation of Data Cleaning and Manipulation

Step 1: Adding New Columns

Objective: Add two new columns, `ride_length` and `day_of_week`, to the dataset to facilitate analysis.

SQL Query:

```
ALTER TABLE projectechnhiesean.BikeCase.2023BikeCase
ADD COLUMN IF NOT EXISTS ride_length FLOAT64,
ADD COLUMN IF NOT EXISTS day_of_week STRING;
```

Explanation:

- `ride_length`: This column stores the duration of each bike ride in minutes.
- `day_of_week`: This column indicates the day of the week each ride started.

Step 2: Populating New Columns

Objective: Populate the `ride_length` and `day_of_week` columns with appropriate data.

SQL Query:

```sql
UPDATE projectechnhiesean.BikeCase.2023BikeCase
SET ride_length = TIMESTAMP_DIFF(ended_at, started_at, MINUTE),
    day_of_week = FORMAT_TIMESTAMP('%A', started_at)
WHERE ride_length IS NULL OR day_of_week IS NULL;
```

Explanation:

- `ride_length` is calculated as the difference in minutes between `ended_at` and `started_at`.
- `day_of_week` is extracted from the `started_at` timestamp using the `FORMAT_TIMESTAMP` function.

Step 3: Identifying Outliers

Objective: Identify and isolate outliers based on ride length and missing data to ensure data quality.

SQL Query:

```sql
CREATE OR REPLACE TABLE projectechnhiesean.BikeCase.Outliers AS
WITH stats AS (
  SELECT
    AVG(ride_length) AS mean_ride_length,
    STDDEV(ride_length) AS stddev_ride_length
  FROM projectechnhiesean.BikeCase.2023BikeCase
),
outliers AS (
  SELECT
    ride_id,
    rideable_type,
    started_at,
    ended_at,
    start_station_name,
    end_station_name,
    start_lat,
    start_lng,
    end_lat,
    end_lng,
    member_casual,
    ride_length,
    day_of_week
  FROM
    projectechnhiesean.BikeCase.2023BikeCase,
    stats
  WHERE
    ride_length > (mean_ride_length + 3 * stddev_ride_length)
    OR ride_length < (mean_ride_length - 3 * stddev_ride_length)
    OR ride_length <= 0
    OR end_station_name IS NULL
    OR end_lat IS NULL
    OR end_lng IS NULL
)
SELECT * FROM outliers;
```

Explanation:

- Outliers are defined as rides with a `ride_length` beyond three standard deviations from the mean or with missing end station data.

Step 4: Cleaning the Original Data Table

Objective: Remove outliers from the original dataset and create a clean dataset.

SQL Query:

```sql
CREATE OR REPLACE TABLE projectechnhiesean.BikeCase.2023BikeCase AS
WITH stats AS (
  SELECT
    AVG(ride_length) AS mean_ride_length,
    STDDEV(ride_length) AS stddev_ride_length
  FROM projectechnhiesean.BikeCase.2023BikeCase
),
cleaned_data AS (
  SELECT
    ride_id,
    rideable_type,
    started_at,
    ended_at,
    start_station_name,
    end_station_name,
    start_lat,
    start_lng,
    end_lat,
    end_lng,
    member_casual,
    ride_length,
    day_of_week
  FROM
    projectechnhiesean.BikeCase.2023BikeCase,
    stats
  WHERE
    ride_length <= (mean_ride_length + 3 * stddev_ride_length)
    AND ride_length >= (mean_ride_length - 3 * stddev_ride_length)
    AND ride_length > 0
    AND end_station_name IS NOT NULL
    AND end_lat IS NOT NULL
    AND end_lng IS NOT NULL
)
SELECT * FROM cleaned_data;
```

Explanation:

- The cleaned dataset excludes outliers to ensure more accurate analysis.

Step 5: Creating a Cleaned Version of the Table with Additional Processing

Objective: Create an enhanced version of the cleaned dataset with additional processing to handle missing station names.

SQL Query:

```sql
CREATE TABLE projectechnhiesean.BikeCase.2023BikeCaseCleaned_V2 AS
SELECT
  ride_id,
  rideable_type,
  started_at,
  ended_at,
  start_station_name,
  end_station_name,
  start_lat,
  start_lng,
  end_lat,
  end_lng,
  member_casual,
  ride_length,
  day_of_week,
  CASE
    WHEN start_station_name IS NULL THEN CONCAT(CAST(start_lat AS STRING), ', ', CAST(start_lng AS STRING))
    ELSE start_station_name
  END AS start_station_combined,
  CASE
    WHEN end_station_name IS NULL THEN CONCAT(CAST(end_lat AS STRING), ', ', CAST(end_lng AS STRING))
    ELSE end_station_name
  END AS end_station_combined
FROM projectechnhiesean.BikeCase.2023BikeCase;
```

Explanation:

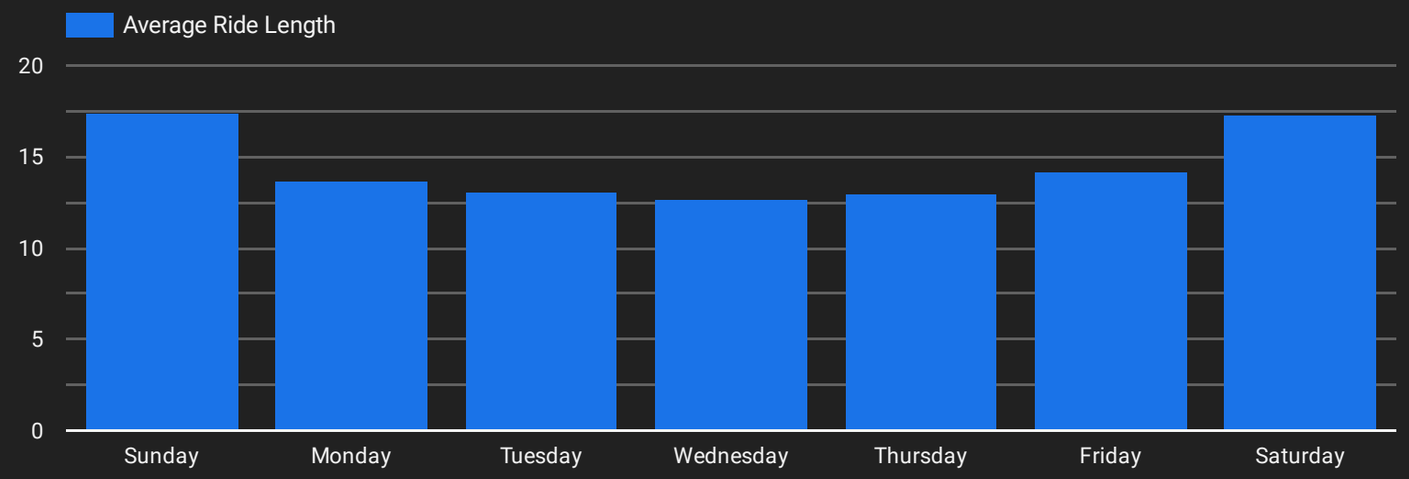
- Combines latitude and longitude coordinates into a single field when station names are missing, ensuring no loss of location information.

2023 Bike Usage Patterns

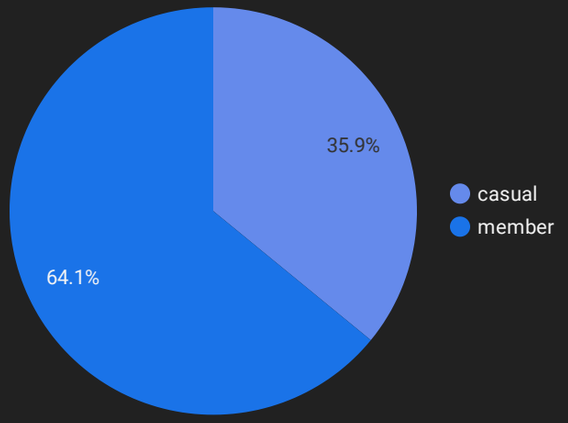
Record Count
5,559,755

Average Ride Length (Minutes)
14.43

Median
9



Count of Rides



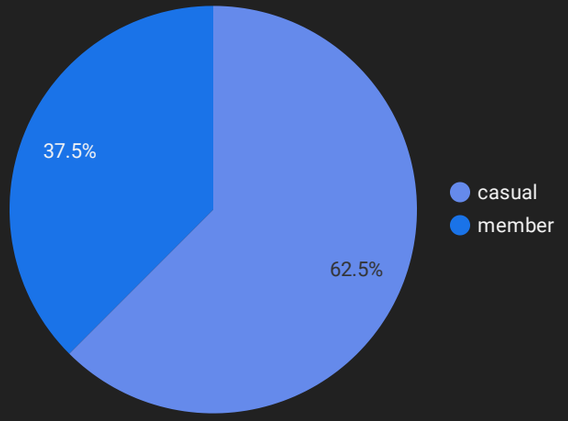
Most Popular Start Stations

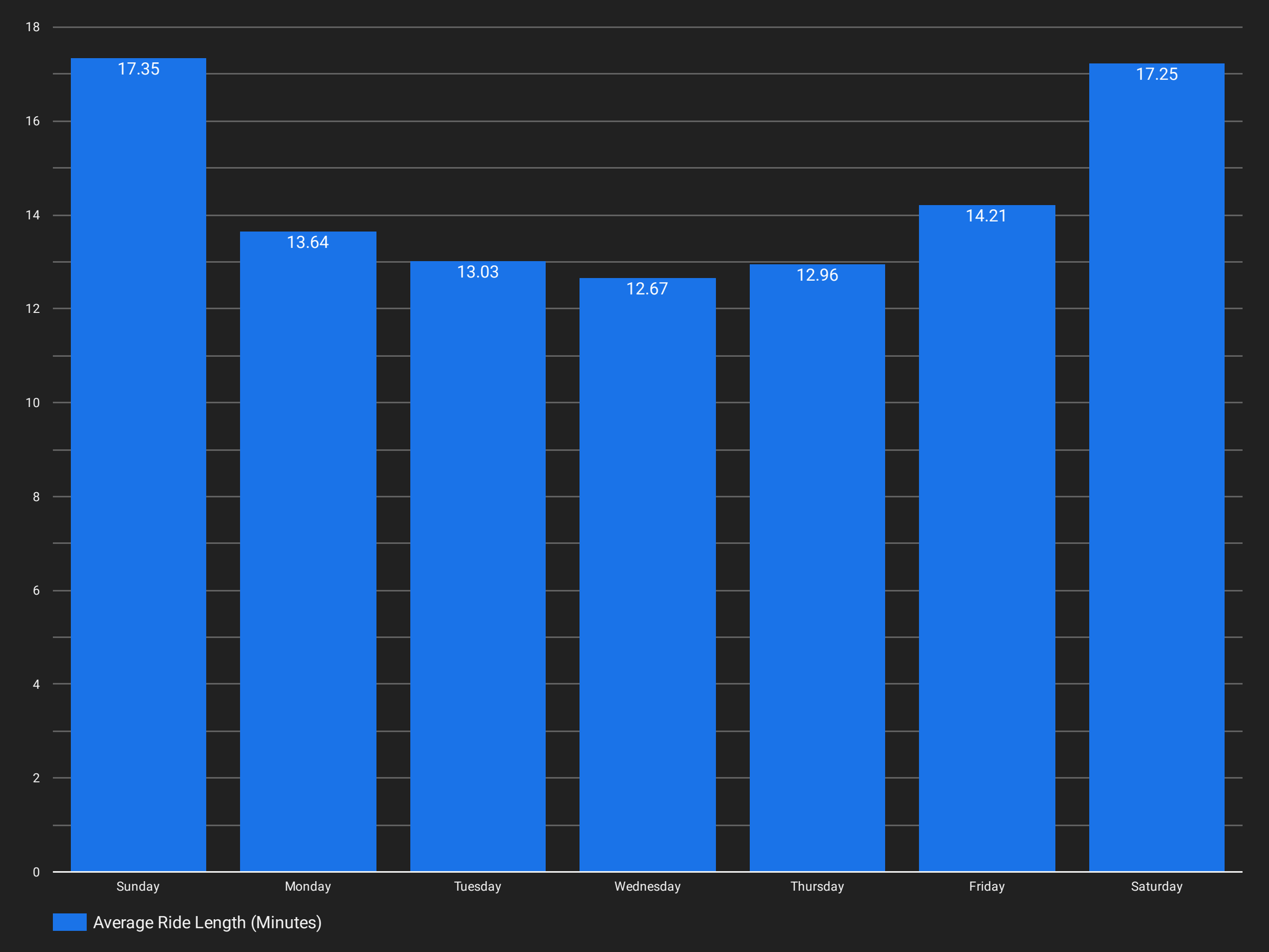
	Station	Record Count
1.	Streeter Dr & Grand Ave	61,617
2.	DuSable Lake Shore Dr & Monroe St	39,256
3.	Michigan Ave & Oak St	36,471
4.	Clark St & Elm St	35,095
5.	DuSable Lake Shore Dr & North Blvd	35,045
6.	Kingsbury St & Kinzie St	34,235
7.	Wells St & Concord Ln	32,918
8.	Clinton St & Washington Blvd	31,951
9.	Wells St & Elm St	29,876
10.	Theater on the Lake	29,365

Most Popular End Stations

	Station	Record Count
1.	Streeter Dr & Grand Ave	62,951
2.	DuSable Lake Shore Dr & North Blvd	38,672
3.	Michigan Ave & Oak St	37,273
4.	DuSable Lake Shore Dr & Monroe St	37,229
5.	Clark St & Elm St	34,352
6.	Kingsbury St & Kinzie St	33,675
7.	Wells St & Concord Ln	33,609
8.	Clinton St & Washington Blvd	32,688
9.	Millennium Park	30,398
10.	Theater on the Lake	30,078

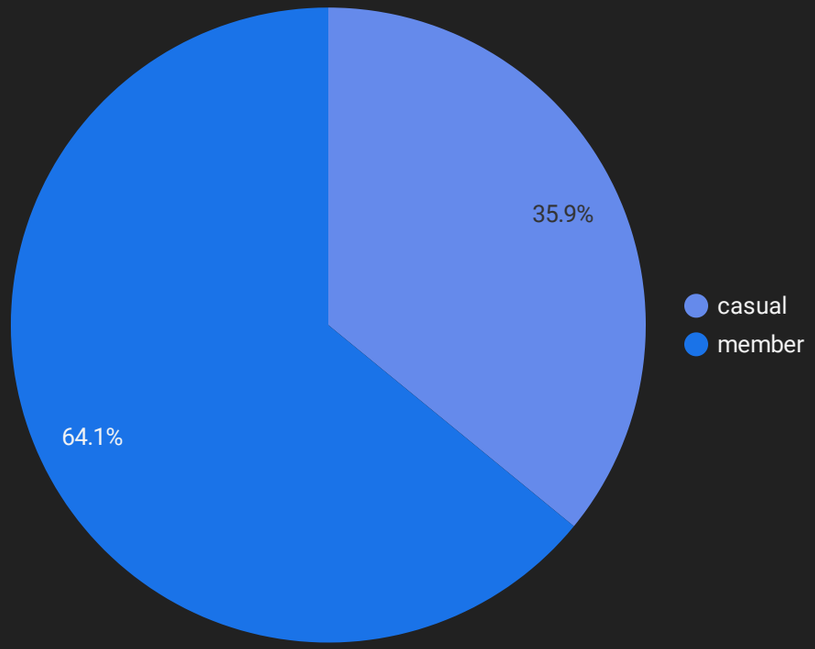
Average Length



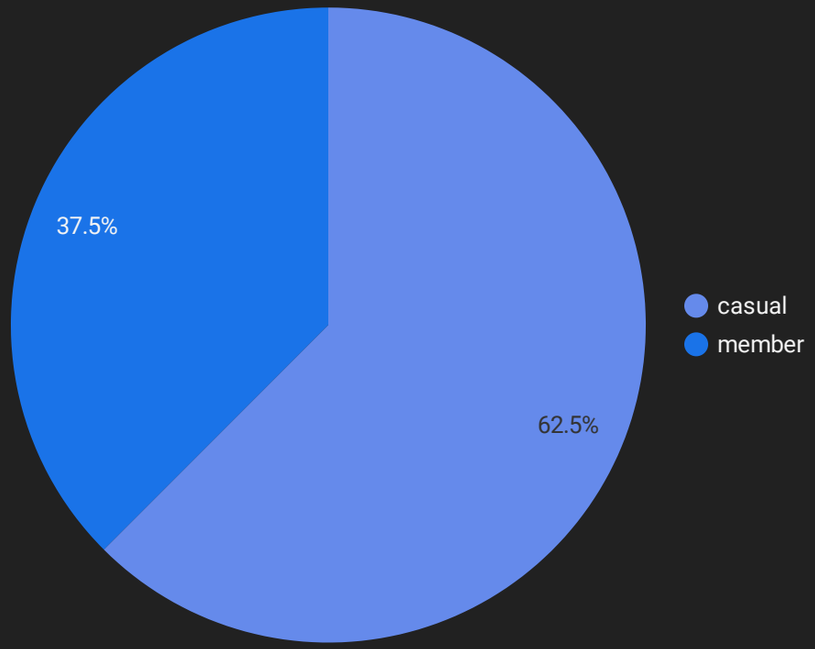

18
16
14
12
10
8
6
4
2
0
17.35
13.64
13.03
12.67
12.96
14.21
17.25
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Average Ride Length (Minutes)

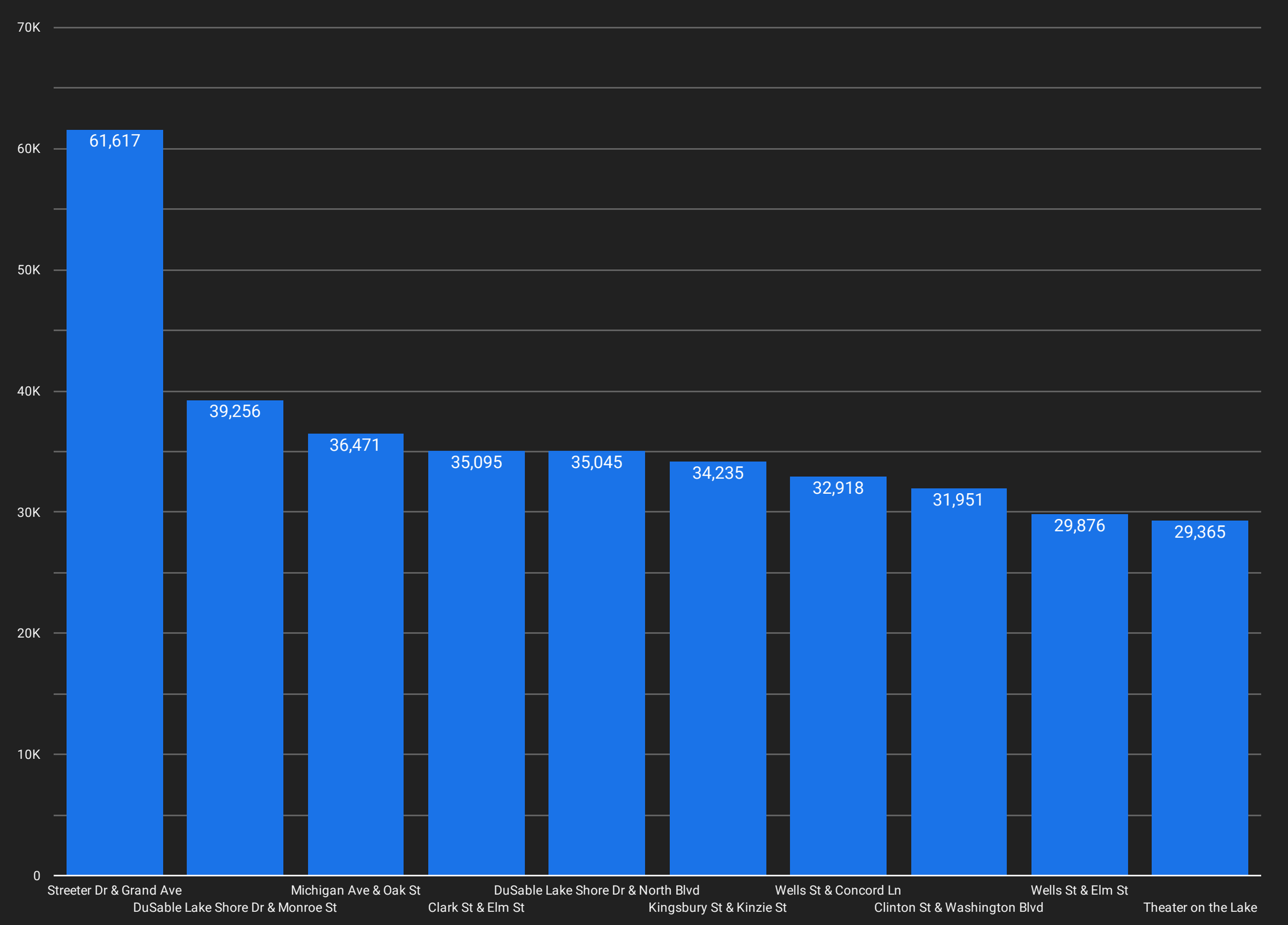
Member VS Casual Comparison

Count of Rides

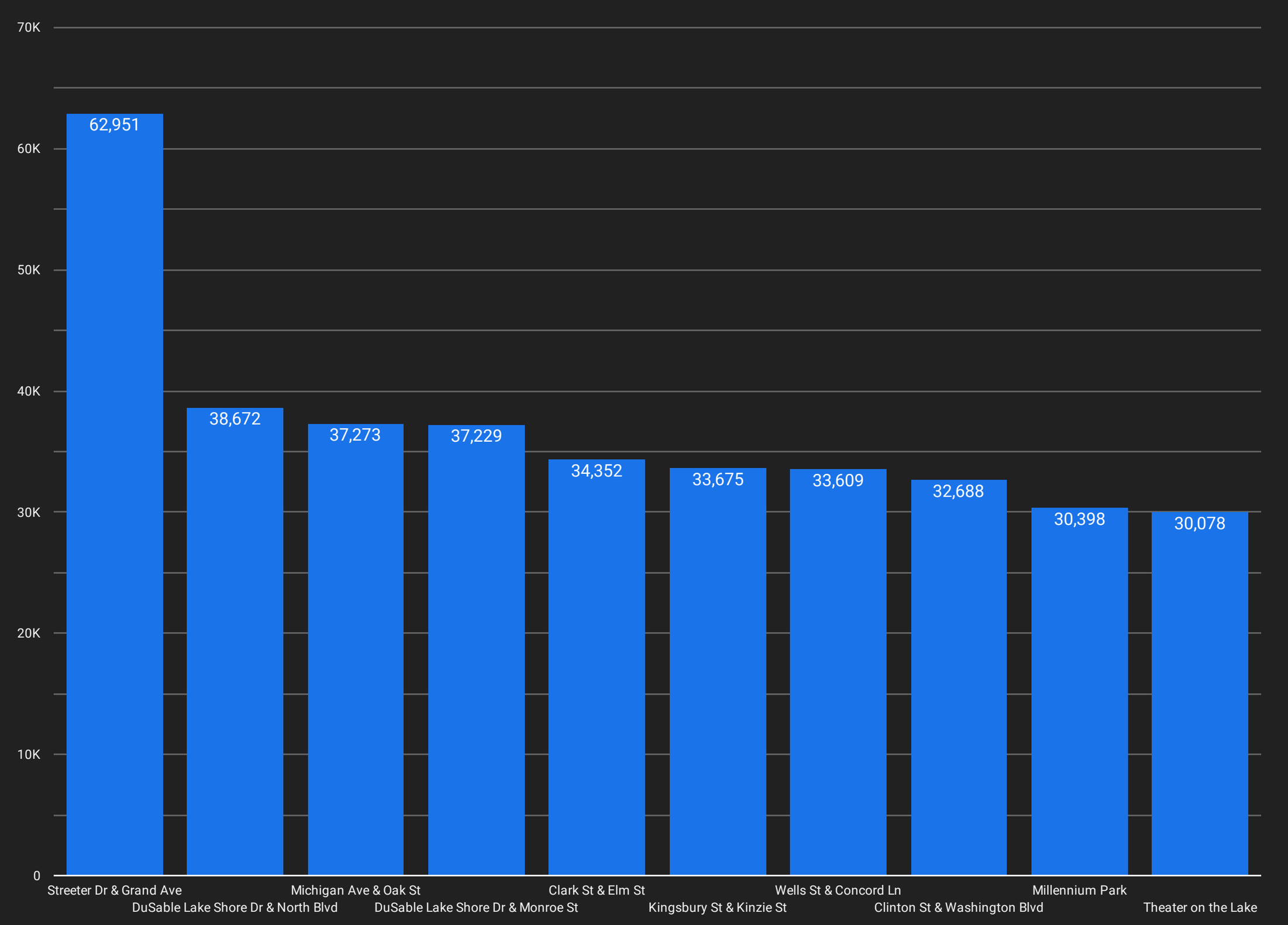


Average Length





Most Popular Start Stations



Most Popular End Stations

Average Ride Length from Start to End Station

	Start	End	Record Count	Minutes
1.	Streeter Dr & Grand Ave	Streeter Dr & Grand Ave	8,915	37.95
2.	Ellis Ave & 60th St	Ellis Ave & 55th St	6,964	4.51
3.	DuSable Lake Shore Dr & Monroe St	DuSable Lake Shore Dr & Monroe St	6,822	34.04
4.	Ellis Ave & 60th St	University Ave & 57th St	6,670	3.97
5.	Ellis Ave & 55th St	Ellis Ave & 60th St	6,399	4.78
6.	University Ave & 57th St	Ellis Ave & 60th St	6,250	4.16
7.	Calumet Ave & 33rd St	State St & 33rd St	5,472	3.93
8.	State St & 33rd St	Calumet Ave & 33rd St	5,374	4.02
9.	DuSable Lake Shore Dr & Monroe St	Streeter Dr & Grand Ave	5,128	25.1
10.	Michigan Ave & Oak St	Michigan Ave & Oak St	4,561	45.7
11.	Loomis St & Lexington St	Morgan St & Polk St	3,748	4.54
12.	Morgan St & Polk St	Loomis St & Lexington St	3,400	4.7
13.	Millennium Park	Millennium Park	3,247	36.76
14.	University Ave & 57th St	Kimbark Ave & 53rd St	3,159	7.17
15.	41.79, -87.6	41.79, -87.6	3,023	4.1
16.	Montrose Harbor	Montrose Harbor	2,962	41.46
17.	Kimbark Ave & 53rd St	University Ave & 57th St	2,919	6.4
18.	Dusable Harbor	Dusable Harbor	2,909	31
19.	Streeter Dr & Grand Ave	DuSable Lake Shore Dr & Monroe St	2,765	23.86
20.	University Ave & 57th St	Lake Park Ave & 56th St	2,656	5.69

Top 3 Recommendations for Converting Casual Riders into Annual Members

1. Implement a "Weekend Warrior" Membership:

- Create a membership tier specifically targeting weekend riders.
- Offer discounted rates for unlimited weekend rides.
- Promote this membership through targeted digital marketing campaigns, highlighting the cost savings for frequent weekend riders.

2. Launch a "Ride More, Save More" Loyalty Program:

- Introduce a points system where casual riders can earn points based on ride duration and frequency.
- Once a certain point threshold is reached, offer a discounted annual membership.
- Use digital media to gamify the experience, showing users their progress towards earning a discounted membership.

3. Develop Partnerships with Local Attractions:

- Collaborate with popular tourist destinations and local businesses near high-traffic stations.
- Create bundled packages that include annual Cyclistic memberships with discounts to partner attractions.
- Use geotargeted digital ads to promote these packages to casual riders when they're near popular stations or attractions.

These recommendations aim to address the different usage patterns of casual riders, incentivize more frequent use, and leverage popular locations to drive membership conversions. Each strategy can be effectively communicated and implemented through digital media channels, aligning with interest in using digital media to influence casual riders.