Case study 1

Objective: Maximize the number of annual memberships

A project-based data analytics life cycle has five simple steps:

1. Identifying the problem
2. Designing data requirements
3. Pre-processing data
4. Performing data analysis
5. Visualizing data

You will produce a report with the following deliverables:

1. A clear statement of the business task

2. A description of all data sources used

3. Documentation of any cleaning or manipulation of data

4. A summary of your analysis

5. Supporting visualizations and key findings

6. Your top three recommendations based on your analysis

# Identifying the question/problem

1. **How do annual members and casual riders use Cyclistic bikes differently?** 
   1. What do they use bikes for? How frequently?
   2. Ride length time
   3. Cities with more members?
   4. Average length time for casuals and members?
   5. Months with higher usage frequency?
   6. Distance

2. Why would casual riders buy Cyclistic annual memberships?

3. How can Cyclistic use digital media to influence casual riders to become members?

a. Look at hystorical data to find correlation between peaks of annual memberships after use of digital media strategies?

Taking the 1st question

## Question: How do annual members and casual riders use Cyclistic bikes differently?

Datasets used are contained [here](https://divvy-tripdata.s3.amazonaws.com/index.html). I used the Divvy datasets concerning Jan 2023 to Dec 2023. Downloaded each one, in .csv file format.

The aim is to join all datasets into one big dataset using SQL queries in BigQuery. I can’t do this in spreadsheets as the final dataset would be too big to be supported by Excel. A worksheet can only have 1,048,576 rows in Microsoft Excel.

# Designing data requirements

* Check for duplicates
* Missing values NA
* Data types
  + Put date time format columns into separate columns

# Data exploration in SQL

Combine all tables in BiqQuery

CREATE TABLE IF NOT EXISTS `2023\_divvy\_tripdata.combined\_data` AS (

  SELECT \*

  FROM `2023\_divvy\_tripdata.202301-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202302-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202303-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202304-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202305-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202306-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202307-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202308-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202309-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202310-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202311-divvy-tripdata`

  UNION ALL

  SELECT \*

  FROM `2023\_divvy\_tripdata.202312-divvy-tripdata`

);

Count number of rows in combined dataset:

SELECT COUNT(\*) FROM `2023\_divvy\_tripdata.combined\_data`;

5719877 rows

Data exploration

* Check for data types

SELECT column\_name, data\_type

FROM `2023\_divvy\_tripdata`.INFORMATION\_SCHEMA.COLUMNS

WHERE table\_name = 'combined\_data';

|  |  |
| --- | --- |
| column\_name | data\_type |
| ride\_id | STRING |
| rideable\_type | STRING |
| started\_at | TIMESTAMP |
| ended\_at | TIMESTAMP |
| start\_station\_name | STRING |
| start\_station\_id | STRING |
| end\_station\_name | STRING |
| end\_station\_id | STRING |
| start\_lat | FLOAT64 |
| start\_lng | FLOAT64 |
| end\_lat | FLOAT64 |
| end\_lng | FLOAT64 |
| member\_casual | STRING |

Data types seem correct

* Check for missing values

SELECT COUNT(\*) - COUNT(ride\_id) as ride\_id,

COUNT(\*) - COUNT(rideable\_type) as rideable\_type,

COUNT(\*) - COUNT(started\_at) as started\_at,

COUNT(\*) - COUNT(ended\_at) as ended\_at,

COUNT(\*) - COUNT(start\_station\_name) as start\_station\_name,

COUNT(\*) - COUNT(end\_station\_name) as end\_station\_name,

COUNT(\*) - COUNT(start\_station\_id) as start\_station\_id,

COUNT(\*) - COUNT(end\_station\_id) as end\_station\_id,

COUNT(\*) - COUNT(start\_lat) as start\_lat,

COUNT(\*) - COUNT(start\_lng) as start\_lng,

COUNT(\*) - COUNT(end\_lat) as end\_lat,

COUNT(\*) - COUNT(end\_lng) as end\_lng,

COUNT(\*) - COUNT(member\_casual) as member\_casual

FROM `2023\_divvy\_tripdata.combined\_data`;

Result:

* Check for duplicate rows

SELECT COUNT(ride\_id) - COUNT(DISTINCT ride\_id)

FROM `2023\_divvy\_tripdata.combined\_data`;

0 duplicate rows

- Checking length of ride\_id column values

SELECT LENGTH(ride\_id) AS length\_ride\_id, COUNT(ride\_id) AS no\_of\_rows

FROM `2023\_divvy\_tripdata.combined\_data`

GROUP BY length\_ride\_id;

Every ride\_id string has length of 16

* Check how many distinct values rideable\_type has

SELECT DISTINCT rideable\_type, COUNT (rideable\_type) as nr\_of\_rows,

FROM `2023\_divvy\_tripdata.combined\_data`

GROUP BY rideable\_type;

Output: 3 different types of bikes

A screenshot of a computer

Description automatically generated

Create columns ride\_length (in min), day\_of\_week, and month while dropping rows with null values:

--Check how many distinct values rideable\_type has

SELECT DISTINCT rideable\_type, COUNT (rideable\_type) as nr\_of\_rows,

FROM `2023\_divvy\_tripdata.combined\_data`

GROUP BY rideable\_type;

--Create columns day\_of\_week, month, trip\_length (in min) & trip\_distance (in m) while dropping rows with null values:

CREATE TABLE IF NOT EXISTS `2023\_divvy\_tripdata.clean\_combined\_data` AS (

  SELECT

    a.ride\_id,

    a.rideable\_type,

    a.started\_at,

    a.ended\_at,

    a.start\_station\_name,

    a.end\_station\_name,

    a.start\_lat,

    a.start\_lng,

    a.end\_lat,

    a.end\_lng,

    a.member\_casual,

    CASE EXTRACT(DAYOFWEEK FROM a.started\_at)

      WHEN 1 THEN 'Sunday'

      WHEN 2 THEN 'Monday'

      WHEN 3 THEN 'Tuesday'

      WHEN 4 THEN 'Wednesday'

      WHEN 5 THEN 'Thursday'

      WHEN 6 THEN 'Friday'

      WHEN 7 THEN 'Saturday'

    END AS day\_of\_week,

    CASE EXTRACT(MONTH FROM a.started\_at)

      WHEN 1 THEN 'January'

      WHEN 2 THEN 'February'

      WHEN 3 THEN 'March'

      WHEN 4 THEN 'April'

      WHEN 5 THEN 'May'

      WHEN 6 THEN 'June'

      WHEN 7 THEN 'July'

      WHEN 8 THEN 'August'

      WHEN 9 THEN 'September'

      WHEN 10 THEN 'October'

      WHEN 11 THEN 'November'

      WHEN 12 THEN 'December'

    END AS month,

    b.trip\_length,

    b.trip\_distance

  FROM `2023\_divvy\_tripdata.combined\_data` AS a

  JOIN (

    SELECT

      ride\_id,

      (EXTRACT(HOUR FROM (ended\_at - started\_at)) \* 60 +

       EXTRACT(MINUTE FROM (ended\_at - started\_at)) +

       EXTRACT(SECOND FROM (ended\_at - started\_at)) / 60) AS trip\_length,

      ST\_DISTANCE(ST\_GEOGPOINT(start\_lng, start\_lat), ST\_GEOGPOINT(end\_lng, end\_lat)) AS trip\_distance

    FROM `2023\_divvy\_tripdata.combined\_data`

  ) AS b

  ON a.ride\_id = b.ride\_id

  WHERE

    a.start\_station\_name IS NOT NULL AND

    a.end\_station\_name IS NOT NULL AND

    a.end\_lat IS NOT NULL AND

    a.end\_lng IS NOT NULL AND

    b.trip\_length > 1 AND

    b.trip\_length < 1440 AND

    b.trip\_distance > 10

);

Output: created a table with 4077286 observations

Set ride\_id as primary key:

ALTER TABLE `2023\_divvy\_tripdata.cleaned\_combined\_data`

ADD PRIMARY KEY(ride\_id) NOT ENFORCED;

Count number of rows:

SELECT COUNT(ride\_id) AS no\_of\_rows

FROM `2023\_divvy\_tripdata.cleaned\_combined\_data`;

# Data analysis

Assess number of casuals and members (2738451 members and 1504981 casuals)

SELECT COUNT(\*) AS nr\_of\_observations,

member\_casual

FROM `2023\_divvy\_tripdata.cleaned\_combined\_data`

GROUP BY member\_casual;

A screenshot of a cell phone

Description automatically generated

Assess number of trips per month between casuals and members

SELECT COUNT(\*) AS nr\_of\_trips,

month,

member\_casual

FROM `2023\_divvy\_tripdata.cleaned\_combined\_data`

GROUP BY month, member\_casual

ORDER BY

CASE

    WHEN month = 'JAN' THEN 1

    WHEN month = 'FEB' THEN 2

    WHEN month = 'MAR' THEN 3

    WHEN month = 'APR' THEN 4

    WHEN month = 'MAY' THEN 5

    WHEN month = 'JUN' THEN 6

    WHEN month = 'JUL' THEN 7

    WHEN month = 'AUG' THEN 8

    WHEN month = 'SEP' THEN 9

    WHEN month = 'OCT' THEN 10

    WHEN month = 'NOV' THEN 11

    WHEN month = 'DEC' THEN 12

  END,

  member\_casual;

Output:

|  |  |  |
| --- | --- | --- |
| nr\_of\_trips | month | member\_casual |
| 27894 | JAN | casual |
| 113657 | JAN | member |
| 30603 | FEB | casual |
| 112094 | FEB | member |
| 43759 | MAR | casual |
| 146650 | MAR | member |
| 101330 | APR | casual |
| 203173 | APR | member |
| 161984 | MAY | casual |
| 274016 | MAY | member |
| 203601 | JUN | casual |
| 303291 | JUN | member |
| 224413 | JUL | casual |
| 315384 | JUL | member |
| 215241 | AUG | casual |
| 336963 | AUG | member |
| 181865 | SEP | casual |
| 297975 | SEP | member |
| 122060 | OCT | casual |
| 264203 | OCT | member |
| 67976 | NOV | casual |
| 196152 | NOV | member |
| 34539 | DEC | casual |
| 125768 | DEC | member |

--assess number of trips per month for casuals

SELECT COUNT(\*) AS nr\_of\_trips,

month,

member\_casual

FROM `2023\_divvy\_tripdata.cleaned\_combined\_data`

WHERE member\_casual = "casual"

GROUP BY month, member\_casual

ORDER BY nr\_of\_trips DESC;

--assess number of trips per month for members

SELECT COUNT(\*) AS nr\_of\_trips,

month,

member\_casual

FROM `2023\_divvy\_tripdata.cleaned\_combined\_data`

WHERE member\_casual = "member"

GROUP BY month, member\_casual

ORDER BY nr\_of\_trips DESC;

Output:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| nr\_of\_trips | month | member\_casual | nr\_of\_trips | month | member\_casual |
| 221590 | July | casual | 335681 | August | member |
| 213237 | August | casual | 313933 | July | member |
| 200805 | June | casual | 301904 | June | member |
| 179994 | September | casual | 296920 | September | member |
| 159603 | May | casual | 272770 | May | member |
| 120960 | October | casual | 263326 | October | member |
| 99809 | April | casual | 202195 | April | member |
| 67397 | November | casual | 195538 | November | member |
| 43228 | March | casual | 146008 | March | member |
| 34196 | December | casual | 125377 | December | member |
| 30219 | February | casual | 113299 | January | member |
| 27627 | January | casual | 111670 | February | member |

Notes:

* Both for casuals and members, the trips are more common in summer months – Jun, Jul, Aug.
* Excluding summer months, the months around summer months, such as Sep and May, have the most trips
* Then come the remaining months with fewer trips. The colder months, like Jan, Feb and Dec, have the lower number of trips. Jan for casuals, Feb for members.
* Members display overall higher number of trips
* There is a big jump of nr of trips for casuals between colder months and warmer months (from Mar to Apr; and even from Feb to Mar)

A screenshot of a cell phone

Description automatically generated--Average ride length (min) for members and casuals

SELECT AVG(ride\_length) as ride\_length,

member\_casual

FROM `2023\_divvy\_tripdata.cleaned\_combined\_data`

GROUP BY member\_casual;

Note: Casuals have longer trips in average than members

-- Determine average trip\_length between casuals and members, per month

SELECT

member\_casual,

AVG(trip\_length) as avg\_trip\_length,

month

FROM `2023\_divvy\_tripdata.clean\_combined\_data`

GROUP BY month,

member\_casual

ORDER BY

avg\_trip\_length DESC,

  CASE

     WHEN month = 'January' THEN 1

    WHEN month = 'February' THEN 2

    WHEN month = 'March' THEN 3

    WHEN month = 'April' THEN 4

    WHEN month = 'May' THEN 5

    WHEN month = 'June' THEN 6

    WHEN month = 'July' THEN 7

    WHEN month = 'August' THEN 8

    WHEN month = 'September' THEN 9

    WHEN month = 'October' THEN 10

    WHEN month = 'November' THEN 11

    WHEN month = 'December' THEN 12

  END,

  member\_casual;

|  |  |  |
| --- | --- | --- |
| member\_casual | avg\_trip\_length | month |
| casual | 23.51593581 | July |
| casual | 22.73445518 | May |
| casual | 22.50717723 | August |
| casual | 22.39293626 | June |
| casual | 22.02990322 | September |
| casual | 20.94688806 | April |
| casual | 20.30614198 | October |
| casual | 16.84690318 | November |
| casual | 16.57315982 | February |
| casual | 15.81050669 | March |
| casual | 15.66655213 | December |
| casual | 14.38135339 | January |
| member | 13.3190469 | July |
| member | 13.20376012 | August |
| member | 12.93490872 | June |
| member | 12.68044158 | May |
| member | 12.59795456 | September |
| member | 11.6576779 | April |
| member | 11.60054793 | October |
| member | 11.01271287 | November |
| member | 10.79926355 | December |
| member | 10.38965971 | February |
| member | 10.19432314 | March |
| member | 10.06870375 | January |

Notes: bigger trips in summer months both for casuals and members, although casuals have bigger trips than members overall

-- Find max trip\_length for casuals and members troughout the year

  SELECT

  member\_casual,

  month,

  MAX(trip\_length) AS max\_trip\_length,

  FROM `2023\_divvy\_tripdata.cleaned\_combined\_data`

  GROUP BY member\_casual,

  month

  ORDER BY

  member\_casual,

  max\_trip\_length DESC;

|  |  |  |
| --- | --- | --- |
| member\_casual | month | max\_trip\_length |
| casual | August | 1437.68 |
| casual | October | 1436.78 |
| casual | July | 1434.05 |
| casual | May | 1431.63 |
| casual | June | 1428.72 |
| casual | March | 1427.57 |
| casual | November | 1426.02 |
| casual | September | 1425.42 |
| casual | February | 1423.62 |
| casual | December | 1414.75 |
| casual | April | 1412.78 |
| casual | January | 1351.22 |
| member | November | 1439.87 |
| member | October | 1439.25 |
| member | August | 1434.43 |
| member | July | 1431.87 |
| member | September | 1429.43 |
| member | April | 1429.4 |
| member | February | 1423.78 |
| member | January | 1422.57 |
| member | March | 1422.37 |
| member | June | 1394.22 |
| member | December | 1368.77 |
| member | May | 1353.13 |

Notes: the max trip length is higher for members (1439.87 min, in November) than casuals (1437.68 in August).

  -- Find min trip\_length for casuals and members troughout the year

  SELECT

  member\_casual,

  month,

  MIN(trip\_length) AS min\_trip\_length,

  FROM `2023\_divvy\_tripdata.cleaned\_combined\_data`

  GROUP BY member\_casual,

  month

  ORDER BY

  member\_casual,

  min\_trip\_length ASC;

Notes: the min trip length is the same (1.02 min) every month, for casuals and members

--Determine average trip\_distance between casuals and members

  SELECT

  member\_casual,

  AVG(trip\_distance) as avg\_trip\_distance

  FROM `2023\_divvy\_tripdata.clean\_combined\_data`

  GROUP BY member\_casual

  ORDER BY avg\_trip\_distance DESC;

A screenshot of a computer

Description automatically generated

-- Determine average trip\_distance between casuals and members, per month

SELECT

member\_casual,

ROUND(AVG(trip\_distance), 2) as avg\_trip\_distance,

month

FROM `2023\_divvy\_tripdata.clean\_combined\_data`

GROUP BY

member\_casual,

month

ORDER BY

member\_casual,

avg\_trip\_distance DESC;

|  |  |  |
| --- | --- | --- |
| member\_casual | avg\_trip\_distance | month |
| casual | 2477.46 | June |
| casual | 2423.83 | May |
| casual | 2416.02 | August |
| casual | 2409.88 | July |
| casual | 2345.48 | September |
| casual | 2285.76 | April |
| casual | 2188.84 | October |
| casual | 2002.11 | November |
| casual | 1934.69 | March |
| casual | 1913.17 | February |
| casual | 1845.27 | December |
| casual | 1767.11 | January |
| member | 2283.27 | July |
| member | 2281.42 | June |
| member | 2245.24 | May |
| member | 2232.84 | August |
| member | 2181.37 | September |
| member | 2116.09 | April |
| member | 2043.36 | October |
| member | 1965.66 | November |
| member | 1906.58 | December |
| member | 1906.55 | March |
| member | 1850.8 | February |
| member | 1796.12 | January |

  -- Determine average trip\_distance between casuals and members, per day of week

SELECT

member\_casual,

ROUND(AVG(trip\_distance), 2) as avg\_trip\_distance,

day\_of\_week

FROM `2023\_divvy\_tripdata.clean\_combined\_data`

GROUP BY

member\_casual,

day\_of\_week

ORDER BY

member\_casual,

avg\_trip\_distance DESC;

|  |  |  |
| --- | --- | --- |
| member\_casual | avg\_trip\_distance | day\_of\_week |
| casual | 2442.84 | Saturday |
| casual | 2417.55 | Sunday |
| casual | 2292.92 | Monday |
| casual | 2278.97 | Friday |
| casual | 2260.36 | Thursday |
| casual | 2213.2 | Tuesday |
| casual | 2183.88 | Wednesday |
| member | 2223.29 | Saturday |
| member | 2205.53 | Sunday |
| member | 2118.31 | Tuesday |
| member | 2115.54 | Thursday |
| member | 2107.92 | Wednesday |
| member | 2079.27 | Friday |
| member | 2062.13 | Monday |

-- Find max trip\_distance for casuals and members troughout the year

  SELECT

  member\_casual,

  month,

  ROUND(MAX(trip\_distance), 2) AS max\_trip\_distance,

  FROM `2023\_divvy\_tripdata.clean\_combined\_data`

  GROUP BY member\_casual,

  month

  ORDER BY

  member\_casual,

  max\_trip\_distance DESC;

|  |  |  |
| --- | --- | --- |
| member\_casual | month | max\_trip\_distance |
| casual | August | 9811358.98 |
| casual | June | 9810778.79 |
| casual | November | 30385.44 |
| casual | September | 29187.23 |
| casual | May | 27672.89 |
| casual | July | 27289.48 |
| casual | October | 25239.62 |
| casual | April | 24831.92 |
| casual | January | 24263.41 |
| casual | February | 22805.25 |
| casual | December | 21909.75 |
| casual | March | 21321.46 |
| member | July | 25796.01 |
| member | May | 25611.02 |
| member | September | 25239.62 |
| member | October | 25239.62 |
| member | June | 23371.69 |
| member | December | 22154.19 |
| member | August | 22086.53 |
| member | February | 21880.18 |
| member | November | 21604.21 |
| member | April | 21150.76 |
| member | January | 20148.53 |
| member | March | 16905.58 |

  -- Find min trip\_distance for casuals and members troughout the year

  SELECT

  member\_casual,

  month,

  ROUND(MIN(trip\_distance), 2) AS min\_trip\_distance,

  FROM `2023\_divvy\_tripdata.clean\_combined\_data`

  GROUP BY member\_casual,

  month

  ORDER BY

  member\_casual,

  min\_trip\_distance ASC;

|  |  |  |
| --- | --- | --- |
| member\_casual | month | min\_trip\_distance |
| casual | June | 10.0 |
| casual | October | 10.0 |
| casual | August | 10.0 |
| casual | September | 10.0 |
| casual | May | 10.0 |
| casual | July | 10.0 |
| casual | January | 10.01 |
| casual | April | 10.01 |
| casual | November | 10.01 |
| casual | March | 10.02 |
| casual | December | 10.04 |
| casual | February | 10.11 |
| member | July | 10.0 |
| member | June | 10.0 |
| member | November | 10.0 |
| member | October | 10.01 |
| member | May | 10.01 |
| member | April | 10.01 |
| member | September | 10.02 |
| member | August | 10.02 |
| member | March | 10.02 |
| member | February | 10.02 |
| member | December | 10.03 |
| member | January | 10.05 |

 -- Find day of week with most trips for casuals and members

  SELECT

  member\_casual,

  day\_of\_week,

  COUNT(\*) as nr\_trips

  FROM `2023\_divvy\_tripdata.clean\_combined\_data`

  GROUP BY

  day\_of\_week,

  member\_casual

  ORDER BY

  member\_casual,

  nr\_trips DESC,

   CASE

    WHEN day\_of\_week = 'Monday' THEN 1

    WHEN day\_of\_week = 'Tuesday' THEN 2

    WHEN day\_of\_week = 'Wednesday' THEN 3

    WHEN day\_of\_week = 'Thursday' THEN 4

    WHEN day\_of\_week = 'Friday' THEN 5

    WHEN day\_of\_week = 'Saturday' THEN 6

    WHEN day\_of\_week = 'Sunday' THEN 7

    END;

|  |  |  |
| --- | --- | --- |
| member\_casual | day\_of\_week | nr\_trips |
| casual | Saturday | 281275 |
| casual | Sunday | 228093 |
| casual | Friday | 209413 |
| casual | Thursday | 184125 |
| casual | Wednesday | 169482 |
| casual | Tuesday | 167131 |
| casual | Monday | 159146 |
| member | Wednesday | 435147 |
| member | Thursday | 434276 |
| member | Tuesday | 431252 |
| member | Friday | 382832 |
| member | Monday | 370235 |
| member | Saturday | 333063 |
| member | Sunday | 291816 |

Notes:

* Sunday is the day with most trips for casuals while Thursday (and very closely Friday) is most preferred for members
* Casuals favor the end of the week (around weekend actually) while members favor the middle of the week – workdays (work commute, grocery shopping?)
* Overall, members have more trips than casuals

-- What time of day do casuals and members prefer to start their trips?

SELECT

member\_casual,

EXTRACT(HOUR FROM started\_at) AS hour,

COUNT(\*) AS nr\_of\_trips

FROM `2023\_divvy\_tripdata.clean\_combined\_data`

WHERE member\_casual = "casual"

GROUP BY

member\_casual,

hour

ORDER BY

member\_casual,

nr\_of\_trips DESC,

hour;

SELECT

member\_casual,

EXTRACT(HOUR FROM started\_at) AS hour,

COUNT(\*) AS nr\_of\_trips

FROM `2023\_divvy\_tripdata.clean\_combined\_data`

WHERE member\_casual = "member"

GROUP BY

member\_casual,

hour

ORDER BY

member\_casual,

nr\_of\_trips DESC,

hour;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| member\_casual | hour | nr\_of\_trips | member\_casual | hour | nr\_of\_trips |
| casual | 17 | 138886 | member | 17 | 291615 |
| casual | 16 | 127022 | member | 16 | 248206 |
| casual | 18 | 116009 | member | 18 | 224300 |
| casual | 15 | 108337 | member | 8 | 188502 |
| casual | 14 | 97671 | member | 15 | 180625 |
| casual | 13 | 94197 | member | 19 | 155944 |
| casual | 12 | 91038 | member | 7 | 153521 |
| casual | 19 | 84106 | member | 14 | 145534 |
| casual | 11 | 77186 | member | 12 | 144778 |
| casual | 10 | 60878 | member | 13 | 143886 |
| casual | 20 | 60053 | member | 11 | 127699 |
| casual | 21 | 49986 | member | 9 | 122117 |
| casual | 8 | 49904 | member | 10 | 108396 |
| casual | 9 | 48398 | member | 20 | 106952 |
| casual | 22 | 44387 | member | 6 | 81890 |
| casual | 7 | 37106 | member | 21 | 81267 |
| casual | 23 | 31671 | member | 22 | 58885 |
| casual | 0 | 22669 | member | 23 | 36483 |
| casual | 6 | 20872 | member | 5 | 25820 |
| casual | 1 | 14524 | member | 0 | 22343 |
| casual | 2 | 8389 | member | 1 | 12796 |
| casual | 5 | 7792 | member | 2 | 6998 |
| casual | 3 | 4263 | member | 4 | 5612 |
| casual | 4 | 3321 | member | 3 | 4452 |

Notes:

* Both casuals and members favor the afternoon time (16h-18h) for their trips
* Members favor rides around morning (7h-8h; going to work?) and afternoon (16h-19h; leaving work?).
* Casuals favor rides from 12h-18h, with particular emphasis in late hours of afternoon (16h-18h).
* What types of bikes do members and casuals prefer?

SELECT

member\_casual,

rideable\_type,

COUNT(\*) AS count

FROM `2023\_divvy\_tripdata.clean\_combined\_data`

GROUP BY

member\_casual,

rideable\_type

ORDER BY

member\_casual,

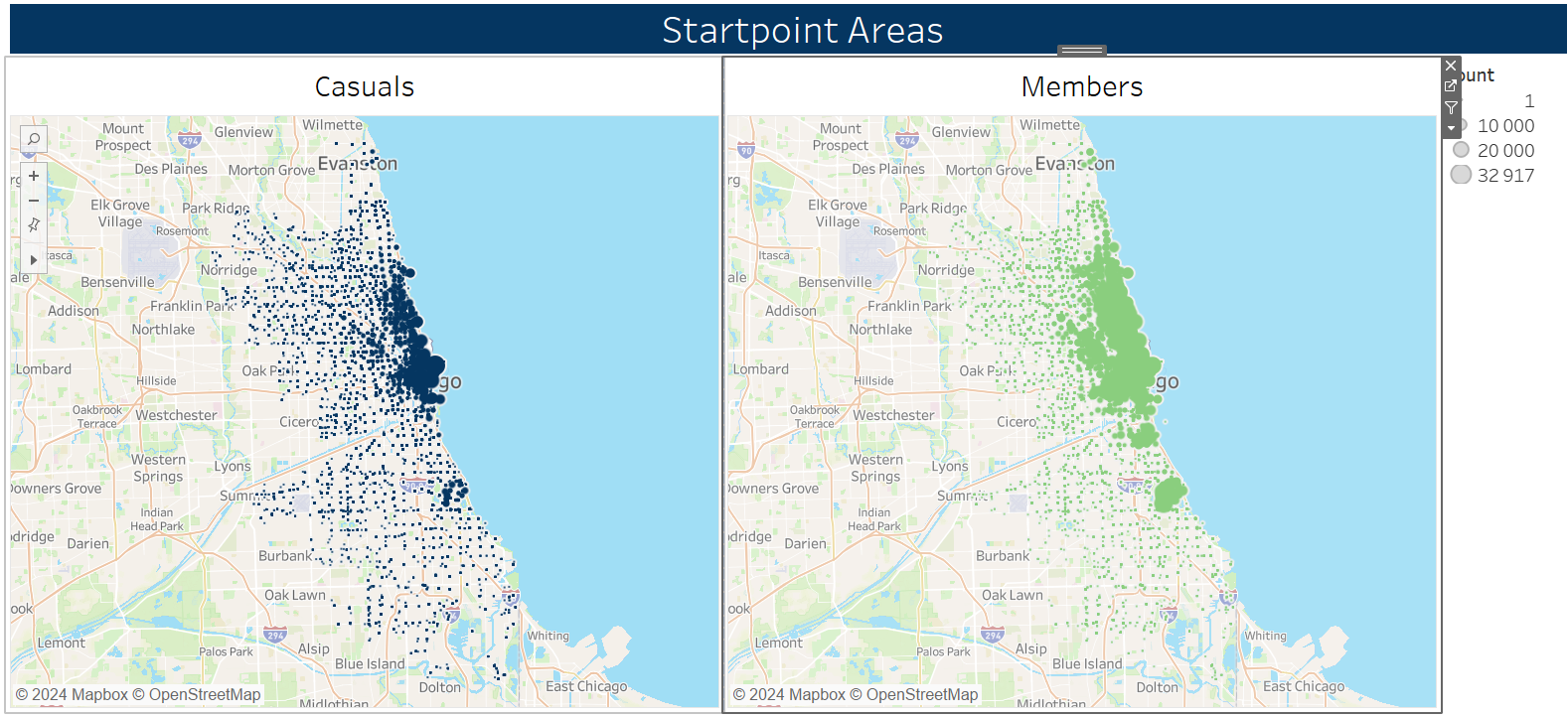
count DESC;

|  |  |  |
| --- | --- | --- |
| member\_casual | rideable\_type | count |
| casual | classic\_bike | 785991 |
| casual | electric\_bike | 551120 |
| casual | docked\_bike | 61554 |
| member | classic\_bike | 1740211 |
| member | electric\_bike | 938410 |

Notes:

* Members prefer using classic bikes (particularly) and electric bikes over docked bikes. Probably for commuting and grocery shopping?
* Casuals favor classic bike and electric bike too but also use docked bike, unlike members. Touristic purposes?

Visualizations Tableau



A screenshot of a graph

Description automatically generated

A graph of a graph

Description automatically generated with medium confidence

A screenshot of a graph

Description automatically generated

A graph of a bar chart

Description automatically generated with medium confidence

A graph of a trip length

Description automatically generated