The company's market department which is led by Lily Moreno would like to increase their revenue and is looking for a way to increase their annual members as he thinks that there is a good chance that casual riders would convert into members as they are already aware of the cyclist program and have chosen cyclist for their transportation. Hence, I have been asked to look into the insight of member and causal riders and analyse how each of these riders uses the services differently and determine what makes casual riders into annual members of the service.

Ask Phase:

The questions that stakeholders would like want answers for are as follows:

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 can be observed from the above questions the most important task is to identify the differences between casual riders and annual member riders and how to use these differences to turn causal riders into annual members that would increase the company's profit.

Prepare:

The data used for this scenario was provided by the Google Analytics course (under this license). The data used for this analysis is collected over the period of 12 months from January 20021 to December 2021. these files were organised in separate files by month and year and saved as a .csv file in compressed zip files.

The data have the following columns/fields:

ride_id — a unique ID per ride
rideable_type: the type of bicycle used
started_at: the date and time that the bicycle was checked out
ended_at: the date and time that the bicycle was checked in
tsart_station_name: the name of the station at the start of the trip
start_station_id: a unique identifier for the start station
end_station_name: the name of the station at the end of the trip
end_station_id : a unique identifier for the end station
Start_lat: the latitude of the start station
start_lng: the longitude of the start station
end_lat: the latitude of the end station
end_lng: the longitude of the end station
member_casual: a field indicating whether the bicycle was taken about by a member or a casual
During the analysis, the following fields were added:

The privacy of users ride-sharing is protected by using the ride_Id which means that there is no personal information of cyclists/riders.

Process Phase:

The tools used during the process phase are:

Excel: Excel is used in the initial cleansing and process of the data where it allowed to quickly and simply transform the data and make a new column for the trip lengths, average ride length and make simple visualisation of which member uses the service more.

RStudio: RStudio is used since it can manage a large amount of data and can-do bulk manipulation, analyse and visualisation.
The first step in R studio was to read all files into a data frame in RStudio and use the string () or glimpse () function to see the types and structure of the data.

```
> str(m12_2021)
spec_tbl_df [247,540 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ ride_id           : chr [1:247540] "46F8167220E4431F" "73A77762838B32FD" "4CF42452054F59C5" "3278BA87B
F698339" ...
 $ rideable_type     : chr [1:247540] "electric_bike" "electric_bike" "electric_bike" "classic_bike" ...
 $ started_at        : POSIXct[1:247540], format: "2021-12-07 15:06:07" "2021-12-11 03:43:29" "2021-12-15
 23:10:28" ...
 $ ended_at          : POSIXct[1:247540], format: "2021-12-07 15:13:42" "2021-12-11 04:10:23" "2021-12-15
 23:23:14" ...
 $ start_station_name: chr [1:247540] "Laflin St & Cullerton St" "LaSalle Dr & Huron St" "Halsted St & No
rth Branch St" "Halsted St & North Branch St" ...
 $ start_station_id  : chr [1:247540] "13307" "KP1705001026" "KA1504000117" "KA1504000117" ...
 $ end_station_name  : chr [1:247540] "Morgan St & Polk St" "Clarendon Ave & Leland Ave" "Broadway & Barr
y Ave" "LaSalle Dr & Huron St" ...
 $ end_station_id    : chr [1:247540] "TA1307000130" "TA1307000119" "13137" "KP1705001026" ...
 $ start_lat         : num [1:247540] 41.9 41.9 41.9 41.9 41.9 ...
 $ start_lng         : num [1:247540] -87.7 -87.6 -87.6 -87.6 -87.7 ...
 $ end_lat           : num [1:247540] 41.9 42 41.9 41.9 41.9 ...
```

Before merging the data frames into one, the number of rows in each data frame s was calculated and saved into total_rows. So that it be used to compare the total number of rows of the data frames after merging all of them.

```
> #Befor merging calculating number of rows
> total_rows<- nrow(m1_2021)+nrow(m2_2021)+nrow(m3_2021)+nrow(m4_2021)+nrow(m5_2021)+nrow(m6_2021)+
+   nrow(m7_2021)+nrow(m8_2021)+nrow(m9_2021)+nrow(m10_2021)+nrow(m11_2021)+nrow(m12_2021)
> # Stack individual quarter's data frames into one big data frame
> cycle_trips <- bind_rows(m1_2021, m2_2021, m3_2021, m4_2021, m5_2021, m6_2021, m7_2021, m8_2021, m9_202
1, m10_2021, m11_2021, m12_2021)
```

Comparing the total number of rows before and after merging.

```
> if(total_rows== nrow(cycle_trips)){
+   print("Equal Number of Rows")
+ } else {
+   print("Error, please check again")
+ }
[1] "Equal Number of Rows"
```

Checking if the data is combined after merging it all.

```
> #Checking if the data is combined
> glimpse(cycle_trips)
Rows: 5,595,063
Columns: 13
$ ride_id            <chr> "E19E6F1B8D4C42ED", "DC88F20C2C55F27F", "EC45C94683FE3F27", "4FA453A75…
$ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", "electric_bike", "e…
$ started_at         <dttm> 2021-01-23 16:14:19, 2021-01-27 18:43:08, 2021-01-21 22:35:54, 2021-0…
$ ended_at           <dttm> 2021-01-23 16:24:44, 2021-01-27 18:47:12, 2021-01-21 22:37:14, 2021-0…
$ start_station_name <chr> "California Ave & Cortez St", "California Ave & Cortez St", "Californi…
$ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660", "17660", "17660", "17660"…
$ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, "Wood St & Augusta Blvd", "Califor…
$ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "657", "13258", "657", "657", "657…
$ start_lat          <dbl> 41.90034, 41.90033, 41.90031, 41.90040, 41.90033, 41.90041, 41.90039, …
$ start_lng          <dbl> -87.69674, -87.69671, -87.69664, -87.69666, -87.69670, -87.69676, -87.…
$ end_lat            <dbl> 41.89000, 41.90000, 41.90000, 41.92000, 41.90000, 41.94000, 41.90000, …
$ end_lng            <dbl> -87.72000, -87.69000, -87.70000, -87.69000, -87.70000, -87.71000, -87.…
$ member_casual      <chr> "member", "member", "member", "member", "casual", "casual", "member", …
```

Checking for null values, as can be seen below some columns have null values.

```
> #Checking for null values
> which(colSums(is.na(cycle_trips))!=0)
start_station_name   start_station_id   end_station_name     end_station_id            end_lat
                 5                  6                  7                  8                 11
           end_lng
                12
```

Removing lat and long columns as this data was removed from the dataset since 2020 according to descriptions.

```
> # Remove lat, long fields as this data was dropped beginning in 2020
> cycle_trips <- cycle_trips %>%
+   select(-c(start_lat, start_lng, end_lat, end_lng))
> # Inspect the new table that has been created
> colnames(cycle_trips)  #List of column names
[1] "ride_id"            "rideable_type"      "started_at"         "ended_at"
[5] "start_station_name" "start_station_id"   "end_station_name"   "end_station_id"
[9] "member_casual"
```

Checking for duplicate values is another step-in data processing since each ride is allocated with a Unique ID. Hence ride_id would be checked for duplicate values. As a been seen below there were no duplicate values in ride_id for cycle_trip data.

```
> #check for duplicate values
> cycle_trips %>%
+   group_by(ride_id) %>%
+   filter(n()>1)
# A tibble: 0 × 13
# Groups:   ride_id [0]
# … with 13 variables: ride_id <chr>, rideable_type <chr>, started_at <dttm>, ended_at <dttm>,
#   start_station_name <chr>, start_station_id <chr>, end_station_name <chr>,
#   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
#   member_casual <chr>
```

After checking for null and duplicate values inspecting the data furthermore

```
> colnames(cycle_trips)  #List of column names
[1] "ride_id"            "rideable_type"      "started_at"         "ended_at"
[5] "start_station_name" "start_station_id"   "end_station_name"   "end_station_id"
[9] "member_casual"
> nrow(cycle_trips)  #How many rows are in data frame?
[1] 5595063
> dim(cycle_trips)  #Dimensions of the data frame?
[1] 5595063       9
> head(cycle_trips)  #See the first 6 rows of data frame.  Also tail(all_trips)
# A tibble: 6 × 9
  ride_id  rideable_type started_at          ended_at            start_station_na… start_station_id
  <chr>    <chr>         <dttm>              <dttm>              <chr>             <chr>
1 E19E6F1… electric_bike 2021-01-23 16:14:19 2021-01-23 16:24:44 California Ave &… 17660
2 DC88F20… electric_bike 2021-01-27 18:43:08 2021-01-27 18:47:12 California Ave &… 17660
3 EC45C94… electric_bike 2021-01-21 22:35:54 2021-01-21 22:37:14 California Ave &… 17660
4 4FA453A… electric_bike 2021-01-07 13:31:13 2021-01-07 13:42:55 California Ave &… 17660
5 BE5E8EB… electric_bike 2021-01-23 02:24:02 2021-01-23 02:24:45 California Ave &… 17660
6 5D8969F… electric_bike 2021-01-09 14:24:07 2021-01-09 15:17:54 California Ave &… 17660
# … with 3 more variables: end_station_name <chr>, end_station_id <chr>, member_casual <chr>
```

```
> str(cycle_trips)  #See list of columns and data types (numeric, character, etc)
tibble [5,595,063 × 9] (S3: tbl_df/tbl/data.frame)
 $ ride_id           : chr [1:5595063] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA453A7
5AE377DB" ...
 $ rideable_type     : chr [1:5595063] "electric_bike" "electric_bike" "electric_bike" "electric_bike"
 ...
 $ started_at        : POSIXct[1:5595063], format: "2021-01-23 16:14:19" "2021-01-27 18:43:08" "2021-01-2
1 22:35:54" ...
 $ ended_at          : POSIXct[1:5595063], format: "2021-01-23 16:24:44" "2021-01-27 18:47:12" "2021-01-2
1 22:37:14" ...
 $ start_station_name: chr [1:5595063] "California Ave & Cortez St" "California Ave & Cortez St" "Califor
nia Ave & Cortez St" "California Ave & Cortez St" ...
 $ start_station_id  : chr [1:5595063] "17660" "17660" "17660" "17660" ...
 $ end_station_name  : chr [1:5595063] NA NA NA NA ...
 $ end_station_id    : chr [1:5595063] NA NA NA NA ...
 $ member_casual     : chr [1:5595063] "member" "member" "member" "member" ...
```

Checking the Statistical summary of data. Mainly for numerical columns.

```
> summary(cycle_trips)  #Statistical summary of data. Mainly for numerics
   ride_id          rideable_type        started_at                    ended_at
 Length:5595063     Length:5595063     Min.   :2021-01-01 00:02:05   Min.   :2021-01-01 00:08:39
 Class :character   Class :character   1st Qu.:2021-06-06 23:52:40   1st Qu.:2021-06-07 00:44:21
 Mode  :character   Mode  :character   Median :2021-08-01 01:52:11   Median :2021-08-01 02:21:55
                                       Mean   :2021-07-29 07:41:02   Mean   :2021-07-29 08:02:58
                                       3rd Qu.:2021-09-24 16:36:16   3rd Qu.:2021-09-24 16:54:05
                                       Max.   :2021-12-31 23:59:48   Max.   :2022-01-03 17:32:18
 start_station_name start_station_id   end_station_name   end_station_id     member_casual
 Length:5595063     Length:5595063     Length:5595063     Length:5595063     Length:5595063
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

Checking the number of observations in each member type column. As seen below casual members are 2529005 and annual members are 3066058.

```
> # Begin by seeing how many observations fall under each usertype
> table(cycle_trips$member_casual)

 casual  member
2529005 3066058
```

Adding columns that list the date, day, month and year for each ride, will allow to aggregate ride data for each month, day, or year. Before completing these operations, we could only aggregate at the ride level.

```
> cycle_trips$date <- as.Date(cycle_trips$started_at) #The default format is yyyy-mm-dd
> cycle_trips$month <- format(as.Date(cycle_trips$date), "%m")
> cycle_trips$day <- format(as.Date(cycle_trips$date), "%d")
> cycle_trips$year <- format(as.Date(cycle_trips$date), "%Y")
> cycle_trips$day_of_week <- format(as.Date(cycle_trips$date), "%A")
```

Calculate the length of each ride by calculating the difference between started_at and ended_at and saving the result into a new column of ride_lengths.

```
> cycle_trips$ride_lengths <- difftime(cycle_trips$ended_at,cycle_trips$started_at)
> # Inspect the structure of the columns
> str(cycle_trips)
```

Converting thee ride_lengths into numeric which allows for calculation on the data

```
is.factor(cycle_trips$ride_lengths)
cycle_trips$ride_lengths <- as.numeric(as.character(cycle_trips$ride_lengths))
is.numeric(cycle_trips$ride_lengths)
```

Removing those entries where thee ride_lengths is negative as some bikes were taken out of the deck and checked for quality by Divvy.

```
> cycle_trips_v2 <- cycle_trips[!(cycle_trips$ride_lengths<0),]
> str(cycle_trips_v2)
tibble [5,594,916 × 15] (S3: tbl_df/tbl/data.frame)
 $ ride_id           : chr [1:5594916] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA453A7
5AE377DB" ...
 $ rideable_type     : chr [1:5594916] "electric_bike" "electric_bike" "electric_bike" "electric_bike"
 ...
 $ started_at        : POSIXct[1:5594916], format: "2021-01-23 16:14:19" "2021-01-27 18:43:08" "2021-01-2
1 22:35:54" ...
 $ ended_at          : POSIXct[1:5594916], format: "2021-01-23 16:24:44" "2021-01-27 18:47:12" "2021-01-2
1 22:37:14" ...
 $ start_station_name: chr [1:5594916] "California Ave & Cortez St" "California Ave & Cortez St" "Califor
nia Ave & Cortez St" "California Ave & Cortez St" ...
 $ start_station_id  : chr [1:5594916] "17660" "17660" "17660" "17660" ...
 $ end_station_name  : chr [1:5594916] NA NA NA NA ...
 $ end_station_id    : chr [1:5594916] NA NA NA NA ...
 $ member_casual     : chr [1:5594916] "member" "member" "member" "member" ...
 $ date              : Date[1:5594916], format: "2021-01-23" "2021-01-27" "2021-01-21" ...
 $ month             : chr [1:5594916] "01" "01" "01" "01" ...
```

Performing descriptive analysis on ride_lengths column as can be seen below, the mean of ride_lengths is 1316.18 seconds the median 720, max 3356649 and minimum is 0.

```
> # Descriptive analysis on ride_length (all figures in seconds)
> mean(cycle_trips_v2$ride_lengths) #straight average (total ride length / rides)
[1] 1316.18
> median(cycle_trips_v2$ride_lengths) #midpoint number in the ascending array of ride lengths
[1] 720
> max(cycle_trips_v2$ride_lengths) #longest ride
[1] 3356649
> min(cycle_trips_v2$ride_lengths) #shortest ride
[1] 0
> # You can condense the four lines above to one line using summary() on the specific attribute
> summary(cycle_trips_v2$ride_lengths)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0     405     720    1316    1307 3356649
```

Running the mean, median, min and max and comparing members and casual users can be observed from below. The casual member seems to be riding more than annual members as can be observed from the mean and median. The maximum length for the causal member is also higher than annual members and both annual and caul members minimum is 0 which should be further looked into.

```
      0     405     720    1316    1307 3356649
> aggregate(cycle_trips_v2$ride_lengths ~ cycle_trips_v2$member_casual, FUN = mean)
  cycle_trips_v2$member_casual cycle_trips_v2$ride_lengths
1                       casual                  1920.1327
2                       member                   818.0129
> aggregate(cycle_trips_v2$ride_lengths ~ cycle_trips_v2$member_casual, FUN = median)
  cycle_trips_v2$member_casual cycle_trips_v2$ride_lengths
1                       casual                         958
2                       member                         576
> aggregate(cycle_trips_v2$ride_lengths ~ cycle_trips_v2$member_casual, FUN = max)
  cycle_trips_v2$member_casual cycle_trips_v2$ride_lengths
1                       casual                     3356649
2                       member                       93596
> aggregate(cycle_trips_v2$ride_lengths ~ cycle_trips_v2$member_casual, FUN = min)
  cycle_trips_v2$member_casual cycle_trips_v2$ride_lengths
1                       casual                           0
2                       member                           0
```

Next, the ride_length on each day of the week is analysed but since the day of the week is out of order, we need to fix it.

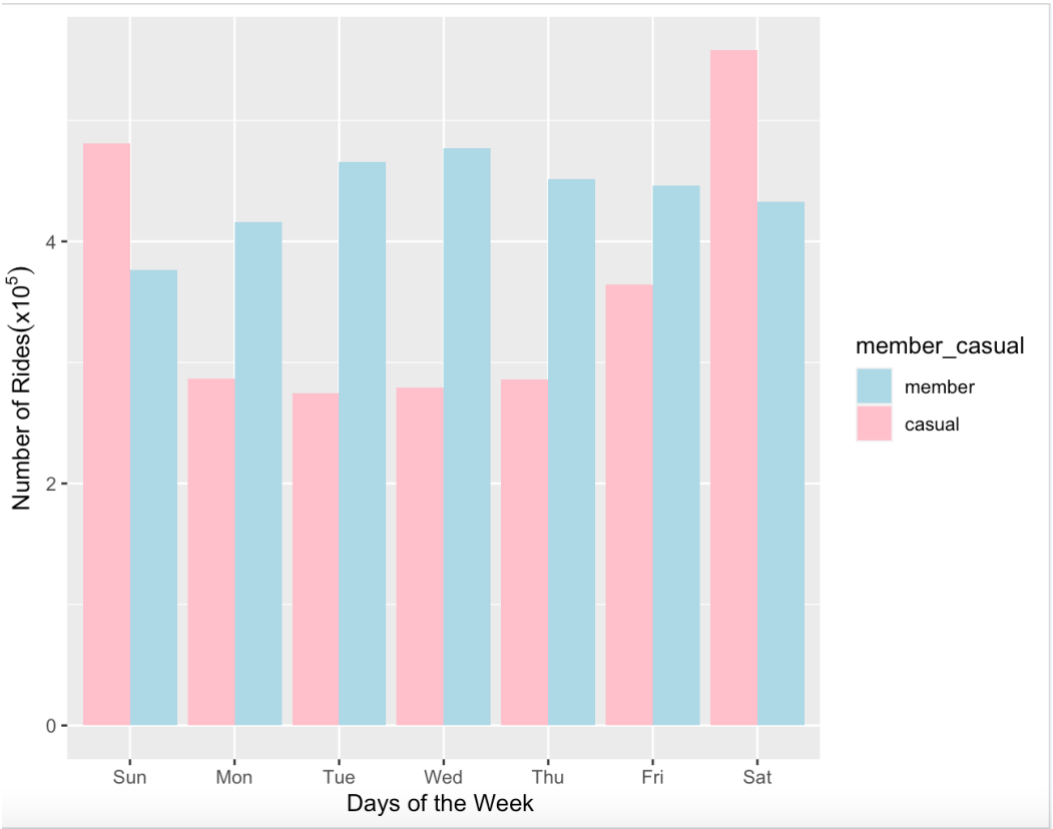
```
> aggregate(cycle_trips_v2$ride_lengths ~ cycle_trips_v2$member_casual + cycle_trips_v2$day_of_week, FUN
 = mean)
   cycle_trips_v2$member_casual cycle_trips_v2$day_of_week cycle_trips_v2$ride_lengths
1                       casual                    Friday                  1820.9160
2                       member                    Friday                   799.4950
3                       casual                    Monday                  1912.5269
4                       member                    Monday                   794.8517
5                       casual                  Saturday                  2082.3740
6                       member                  Saturday                   915.8742
7                       casual                    Sunday                  2253.9949
8                       member                    Sunday                   939.4763
9                       casual                  Thursday                  1662.1955
10                      member                  Thursday                   766.5710
11                      casual                   Tuesday                  1678.3396
12                      member                   Tuesday                   767.2874
13                      casual                 Wednesday                  1659.4383
14                      member                 Wednesday                   769.1496
```

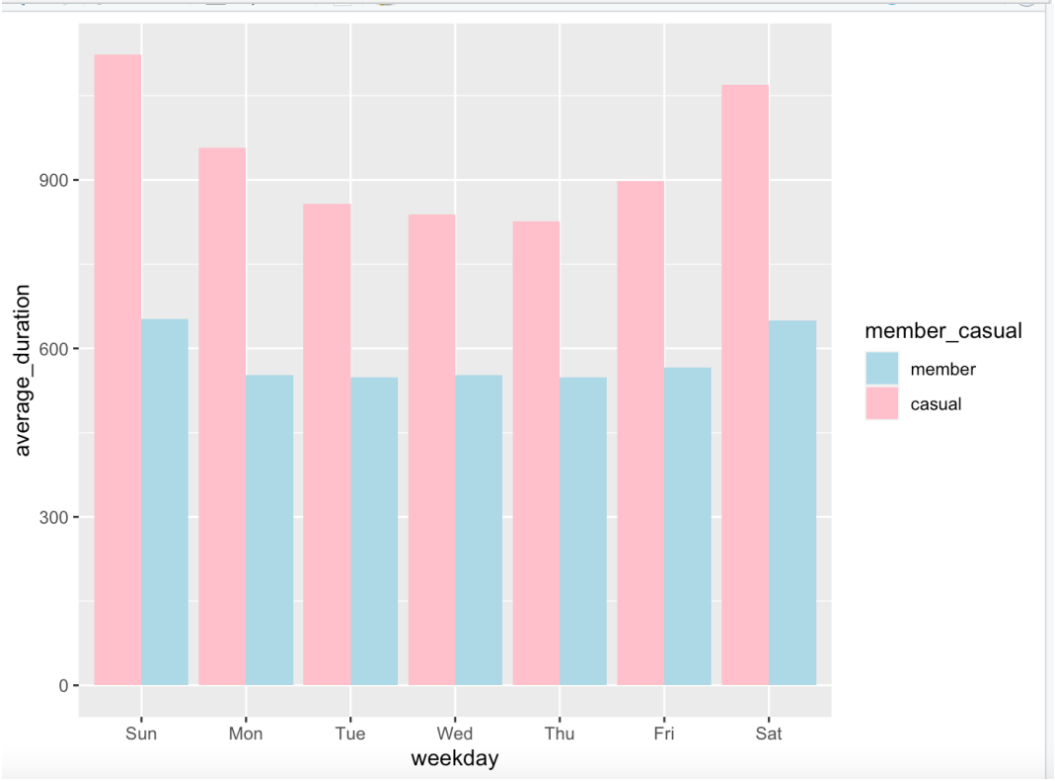
After fixing the order of days of the week, casual members, in general, have a higher length of ride as compared to annual members. Visualising this would help us to understand it better.

```
> aggregate(cycle_trips_v2$ride_lengths ~ cycle_trips_v2$member_casual + cycle_trips_v2$day_of_week, FUN
 = mean)
   cycle_trips_v2$member_casual cycle_trips_v2$day_of_week cycle_trips_v2$ride_lengths
1                       casual                    Sunday                  2253.9949
2                       member                    Sunday                   939.4763
3                       casual                    Monday                  1912.5269
4                       member                    Monday                   794.8517
5                       casual                   Tuesday                  1678.3396
6                       member                   Tuesday                   767.2874
7                       casual                 Wednesday                  1659.4383
8                       member                 Wednesday                   769.1496
9                       casual                  Thursday                  1662.1955
10                      member                  Thursday                   766.5710
11                      casual                    Friday                  1820.9160
12                      member                    Friday                   799.4950
13                      casual                  Saturday                  2082.3740
14                      member                  Saturday                   915.8742
>
```

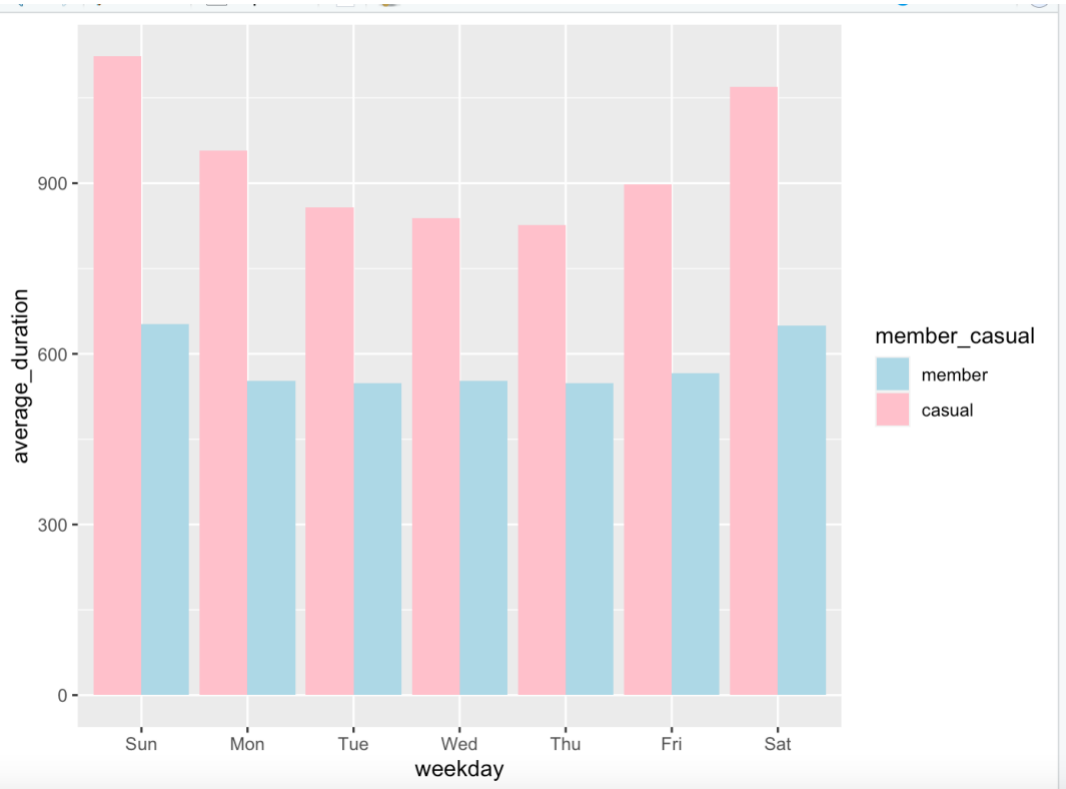
As shown in the graph the number of rides of causal members are higher than annual members. With Saturday and Sunday being the highest for casual members this proves shows that this increase can be due to these two days being weekend and casual member use for recreational purposes while for annual members this seems to be the opposite as during these two days the number of

rides for annual members to decrease while it again increases during the weekdays, with Tuesday and Wednesday holding the highest number of rides for annual members.

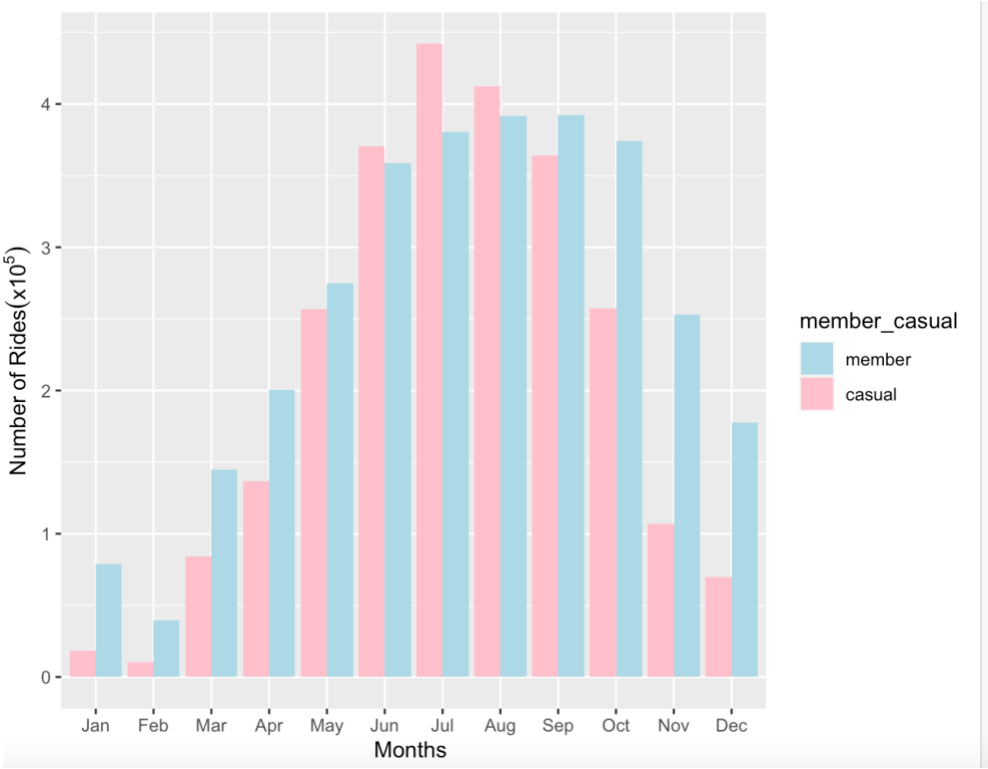




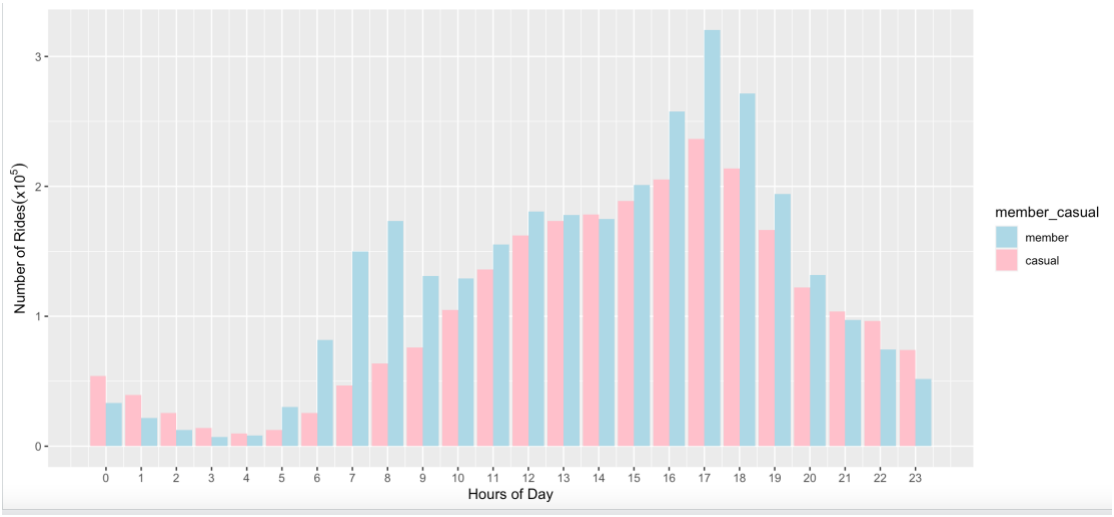
The average duration of these rides also further points towards the difference between casual and annual members with annual members having constant ride duration during weekdays and for causal members the ride duration increase during the weekend (Saturday and Sunday).



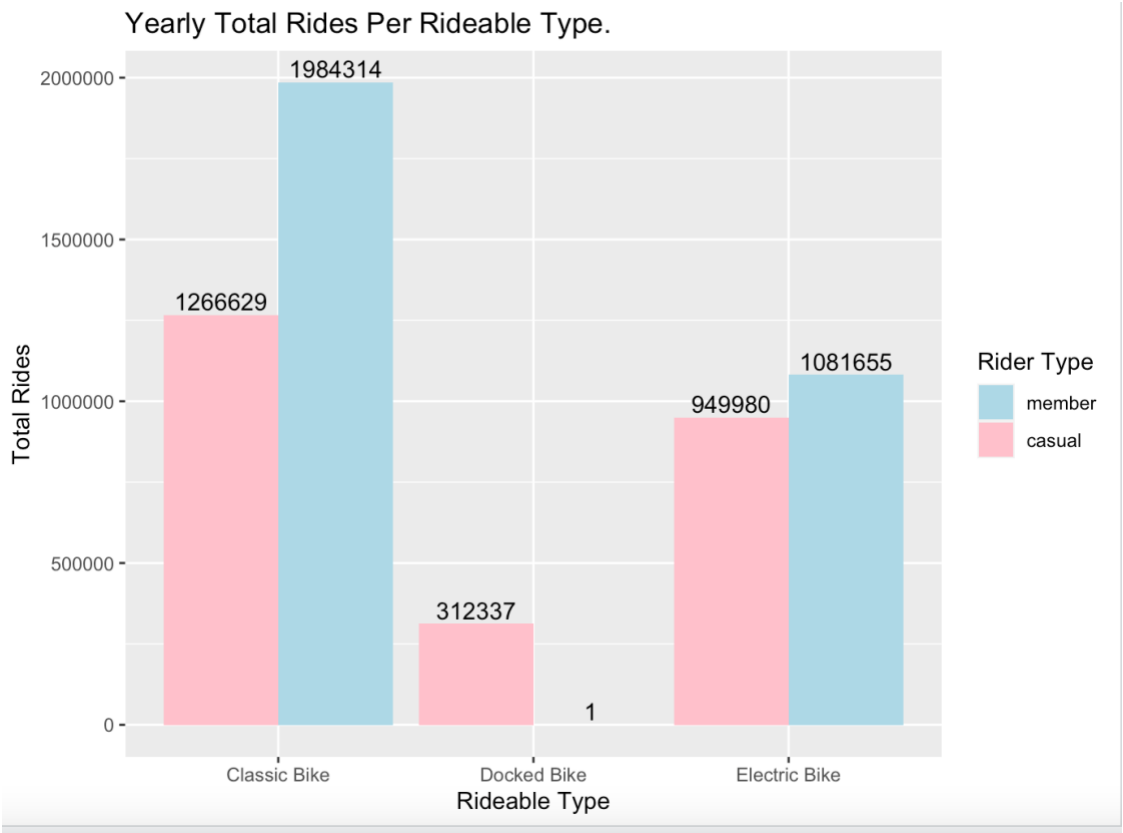
The visualisation is further analysed and as shown in the graph below, the number of rides for both casual members and annual members do decrease during the cold months/seasons especially for the casual member who uses it for recreational purpose but it starts to increase again during warmer months/seasons. For annual members, the decrease is not as extreme as compared to casual members.



The use of cycles rises from 6 am to 8 am for annual members and again from 16 to 19 which are the rush hours for getting to work and getting off the work which further confirms our theory that annual members use the cycle for commencing to their work and off their work.



And finally, which bike type is popular amongst both casual and annual members. It looks like the classic bike is more popular amongst both casual and annual members.



Act:

As per analyses done, it is clear that casual and annual members use the services for different purposes. Casual members use the rides/services for recreational purposes while the annual members use the rides for commuting to and from their workplaces or their regular destination. This is further proved by the fact that annual members use of rides decreased during the weekend while the opposite was concluded for the causal member as their use of rides increase during the weekend (Saturday and Sunday). This is also confirmed by the hourly use of these rides, the peak times/hours for the use of rides were during the commencing and finishing off the work times for annual members.
The ride duration for annual members is almost constant during the weekdays but decreases during the weekend. As for casual members, it increases during the weekend while for annual members it Whether also affects the use of rides/bicycles as in the cold months the number of rides decreases for both casual and annual members. Whereas in warm months the number of rides increases for both annual members and casual riders the increase is significantly higher than those of annual members.

Recommendation:

If the company wants to convert casual riders to members, then they can make a new membership plan where they can offer flexible prices as compared to their regular prices for casual riders during the weekend.

Another thing they can do is do a campaign targeting casual riders who use the services for a long ride and offer them a discount.

Since the rideshare, services are popular during the warmer season, it's beneficial to advertise their services during these seasons.

They can make use of social media to get more insight into their users' likes and dislikes and ask them for suggestions through polls and giveaways.