Bikeshare data Analysis Capstone

Budhiman Dang

2022-07-21

#DATA PREPARATION PHASE

##Setting up my environment Notes:Setting up my Environment with tidyverse, skimr, janitor and dplyr packages

```
library("tidyverse")
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6     ✔ purrr   0.3.4
## ✔ tibble  3.1.7     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library("dplyr")
library("skimr")
library("janitor")
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
setwd("C:/Users/91828/Documents/BDANG/4282234/excel data")
march_csv<-read_csv("march_2020.csv")
```

```
## Rows: 228496 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
april_csv<-read_csv("april_2020.csv")
```

```
## Rows: 84776 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
may_csv<-read_csv("may_2020.csv")
```

```
## Rows: 200274 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jun_csv<-read_csv("jun_2020.csv")
```

```
## Rows: 343005 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jul_csv<-read_csv("jul_2020.csv")
```

```
## Rows: 551480 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
aug_csv<-read_csv("aug_2020.csv")
```

```
## Rows: 622361 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
sep_csv<-read_csv("sep_2020.csv")
```

```
## Rows: 532958 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
oct_csv<-read_csv("oct_2020.csv")
```

```
## Rows: 388653 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
nov_csv<-read_csv("nov_2020.csv")
```

```
## Rows: 259716 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dec_csv<-read_csv("dec_2020.csv")
```

```
## Rows: 131573 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jan21_csv<-read_csv("jan_2021.csv")
```

```
## Rows: 96834 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
feb21_csv<-read_csv("feb_2021.csv")
```

```
## Rows: 49622 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
─
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
compare_df_cols(march_csv,april_csv,may_csv,jun_csv,jul_csv,aug_csv,sep_csv,oct_csv,nov_csv,dec_csv,jan21_csv,feb21_csv,return="mismatch")
```

```
##       column_name march_csv april_csv may_csv jun_csv jul_csv aug_csv sep_csv
## 1   end_station_id character   numeric numeric numeric numeric numeric numeric
## 2 start_station_id character   numeric numeric numeric numeric numeric numeric
##   oct_csv nov_csv   dec_csv jan21_csv feb21_csv
## 1 numeric numeric character character character
## 2 numeric numeric character character character
```

```
march_csv<-mutate(march_csv,end_station_id=as.numeric(end_station_id),start_station_id=as.numeric(start_station_id))
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
dec_csv<-mutate(dec_csv,end_station_id=as.numeric(end_station_id),start_station_id=as.numeric(start_station_id))
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
jan21_csv<-mutate(jan21_csv,end_station_id=as.numeric(end_station_id),start_station_id=as.numeric(start_station_id))
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
feb21_csv<-mutate(feb21_csv,end_station_id=as.numeric(end_station_id),start_station_id=as.numeric(start_station_id))
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion

## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

```
compare_df_cols(march_csv,april_csv,may_csv,jun_csv,jul_csv,aug_csv,sep_csv,oct_csv,nov_csv,dec_csv,jan21_csv,feb21_csv,return="mismatch")
```

```
##  [1] column_name march_csv   april_csv   may_csv     jun_csv     jul_csv    
##  [7] aug_csv     sep_csv     oct_csv     nov_csv     dec_csv     jan21_csv  
## [13] feb21_csv  
## <0 rows> (or 0-length row.names)
```

```
compiled_ride_data_unclean<-rbind(march_csv,april_csv,may_csv,jun_csv,jul_csv,aug_csv,sep_csv,oct_csv,nov_csv,dec_csv,jan21_csv,feb21_csv)
```

```
compiled_ride_data_clean<-compiled_ride_data_unclean %>%
  select(-c(start_lat, start_lng, end_lat, end_lng))
```

```
colnames(compiled_ride_data_clean)
```

```
## [1] "ride_id"            "rideable_type"      "started_at"        
## [4] "ended_at"           "start_station_name" "start_station_id"  
## [7] "end_station_name"   "end_station_id"     "member_casual"
```

```
compiled_ride_data_clean<-compiled_ride_data_clean %>%
  rename(bike_type=rideable_type,
         start_time=started_at,
         end_time=ended_at,
         from_station_name=start_station_name,
         to_station_name=end_station_name,
         from_station_id=start_station_id,
         to_station_id=end_station_id,
         rider_type=member_casual)
```

```
colnames(compiled_ride_data_clean)
```

```
## [1] "ride_id"           "bike_type"         "start_time"       
## [4] "end_time"          "from_station_name" "from_station_id"  
## [7] "to_station_name"   "to_station_id"     "rider_type"
```

```
dim(compiled_ride_data_clean)
```

```
## [1] 3489748       9
```

```
head(compiled_ride_data_clean)
```

```
## # A tibble: 6 × 9
##   ride_id        bike_…¹ start_time          end_time            from_…² from_…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>     <dbl>
## 1 CFA86D4455AA1… classi… 2021-03-16 08:32:30 2021-03-16 08:36:34 Humbol…   15651
## 2 30D9DC61227D1… classi… 2021-03-28 01:26:28 2021-03-28 01:36:55 Humbol…   15651
## 3 846D87A15682A… classi… 2021-03-11 21:17:29 2021-03-11 21:33:53 Shield…   15443
## 4 994D05AA75A16… classi… 2021-03-11 13:26:42 2021-03-11 13:55:41 Winthr…      NA
## 5 DF7464FBE92D8… classi… 2021-03-21 09:09:37 2021-03-21 09:27:33 Glenwo…     525
## 6 CEBA8516FD17F… classi… 2021-03-20 11:08:47 2021-03-20 11:29:39 Glenwo…     525
## # … with 3 more variables: to_station_name <chr>, to_station_id <dbl>,
## #   rider_type <chr>, and abbreviated variable names ¹bike_type,
## #   ²from_station_name, ³from_station_id
## # ℹ Use `colnames()` to see all variable names
```

```
str(compiled_ride_data_clean)
```

```
## tibble [3,489,748 × 9] (S3: tbl_df/tbl/data.frame)
##  $ ride_id          : chr [1:3489748] "CFA86D4455AA1030" "30D9DC61227D1AF3" "846D87A15682A284" "994D05AA75A168F2" ...
##  $ bike_type        : chr [1:3489748] "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ start_time       : POSIXct[1:3489748], format: "2021-03-16 08:32:30" "2021-03-28 01:26:28" ...
##  $ end_time         : POSIXct[1:3489748], format: "2021-03-16 08:36:34" "2021-03-28 01:36:55" ...
##  $ from_station_name: chr [1:3489748] "Humboldt Blvd & Armitage Ave" "Humboldt Blvd & Armitage Ave" "Shields Ave & 28th Pl" "Winthrop Ave & Lawrence Ave" ...
##  $ from_station_id  : num [1:3489748] 15651 15651 15443 NA 525 ...
##  $ to_station_name  : chr [1:3489748] "Stave St & Armitage Ave" "Central Park Ave & Bloomingdale Ave" "Halsted St & 35th St" "Broadway & Sheridan Rd" ...
##  $ to_station_id    : num [1:3489748] 13266 18017 NA 13323 NA ...
##  $ rider_type       : chr [1:3489748] "casual" "casual" "casual" "casual" ...
```

```
summary(compiled_ride_data_clean)
```

```
##    ride_id           bike_type           start_time                    
##  Length:3489748     Length:3489748     Min.   :2020-04-01 00:00:30.00  
##  Class :character   Class :character   1st Qu.:2020-07-14 19:38:28.00  
##  Mode  :character   Mode  :character   Median :2020-08-29 14:50:36.50  
##                                        Mean   :2020-09-10 01:21:45.98  
##                                        3rd Qu.:2020-10-20 18:14:13.00  
##                                        Max.   :2021-03-31 23:59:08.00  
##                                                                        
##     end_time                      from_station_name  from_station_id 
##  Min.   :2020-04-01 00:10:45.00   Length:3489748     Min.   :    2   
##  1st Qu.:2020-07-14 20:13:07.75   Class :character   1st Qu.:  109   
##  Median :2020-08-29 15:21:13.00   Mode  :character   Median :  212   
##  Mean   :2020-09-10 01:46:31.98                      Mean   : 1016   
##  3rd Qu.:2020-10-20 18:28:46.25                      3rd Qu.:  332   
##  Max.   :2021-04-06 11:00:11.00                      Max.   :20258   
##                                                      NA's   :388463  
##  to_station_name    to_station_id     rider_type       
##  Length:3489748     Min.   :    2   Length:3489748    
##  Class :character   1st Qu.:  110   Class :character  
##  Mode  :character   Median :  213   Mode  :character  
##                     Mean   : 1016                     
##                     3rd Qu.:  332                     
##                     Max.   :20258                     
##                     NA's   :404359                    
```

```
skim(compiled_ride_data_clean)
```

Data summary

Name	compiled_ride_data_clean
Number of rows	3489748
Number of columns	9
________________________	
Column type frequency:	
character	5
numeric	2
POSIXct	2

________________________	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1.00	16	16	0	3489539	0
bike_type	0	1.00	11	13	0	3	0
from_station_name	122175	0.96	10	53	0	708	0
to_station_name	143242	0.96	10	53	0	706	0
rider_type	0	1.00	6	6	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
from_station_id	388463	0.89	1016.15	3178.51	2	109	212	332	20258	▇▁▁▁▁
to_station_id	404359	0.88	1015.94	3175.25	2	110	213	332	20258	▇▁▁▁▁

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
start_time	0	1	2020-04-01 00:00:30	2021-03-31 23:59:08	2020-08-29 14:50:36	3040228
end_time	0	1	2020-04-01 00:10:45	2021-04-06 11:00:11	2020-08-29 15:21:13	3027775

```
compiled_ride_data_clean$date<-as.Date(compiled_ride_data_clean$start_time)
compiled_ride_data_clean$year<-format(as.Date(compiled_ride_data_clean$date),"%Y")
compiled_ride_data_clean$month<-format(as.Date(compiled_ride_data_clean$date),"%m")
compiled_ride_data_clean$day<-format(as.Date(compiled_ride_data_clean$date),"%d")
compiled_ride_data_clean$day_of_week<-format(as.Date(compiled_ride_data_clean$date),"%A")
```

```
compiled_ride_data_clean$ride_length<-difftime(compiled_ride_data_clean$end_time,compiled_ride_data_clean$start_time)
```

```
#is.numeric(compiled_ride_data_clean$ride_length)
#is.factor(compiled_ride_data_clean$ride_length)
compiled_ride_data_clean$ride_length<-as.numeric(as.character(compiled_ride_data_clean$ride_length))
```

```
compiled_ride_data_clean_new<-compiled_ride_data_clean[!compiled_ride_data_clean$ride_length<0,]
```

```
summary(compiled_ride_data_clean_new$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     476     874    1677    1601 3523202
```

```
write.csv(compiled_ride_data_clean_new,"merge_clean_data.csv")
```

#DATA ANALYSIS PHASE ###Note:-We will be analysing the data both with R and EXCEL

```
data_clean<-read.csv("merge_clean_data.csv")
rider_type_duration<-data_clean%>%
  group_by(rider_type)%>%
  summarise_at(vars(ride_length),
               list(name=sum))
```

```
casual_rider_duration<-rider_type_duration%>%
  subset(rider_type=="casual")%>%
  select(name)

member_rider_duration<-rider_type_duration%>%
  subset(rider_type=="member")%>%
  select(name)
```

```
casual_member_relationship<-(casual_rider_duration/member_rider_duration)
View(casual_member_relationship)
```

##The above result shows that casual rider duration is 1.94 times than member rider duration.

```
day_rider_counts<-data_clean%>%
  count(day_of_week)
max_val<-max(day_rider_counts$n)
max_rider_day<-day_rider_counts%>%
  subset(day_rider_counts$n==max_val)
head(max_rider_day)
```

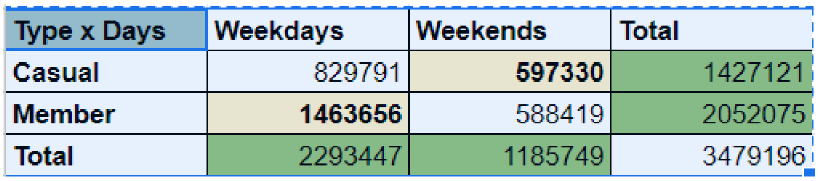
```
##   day_of_week      n
## 3    Saturday 658179
```

```
#Saturday is the pick day
min_val<-min(day_rider_counts$n)
min_rider_day<-day_rider_counts%>%
  subset(day_rider_counts$n==min_val)
head(min_rider_day)
```

```
##   day_of_week      n
## 2      Monday 418505
```

```
#Monday is the minimum rush day
```

#Checking the relation between Weekday and Weekend rider counts by rider_type using excel

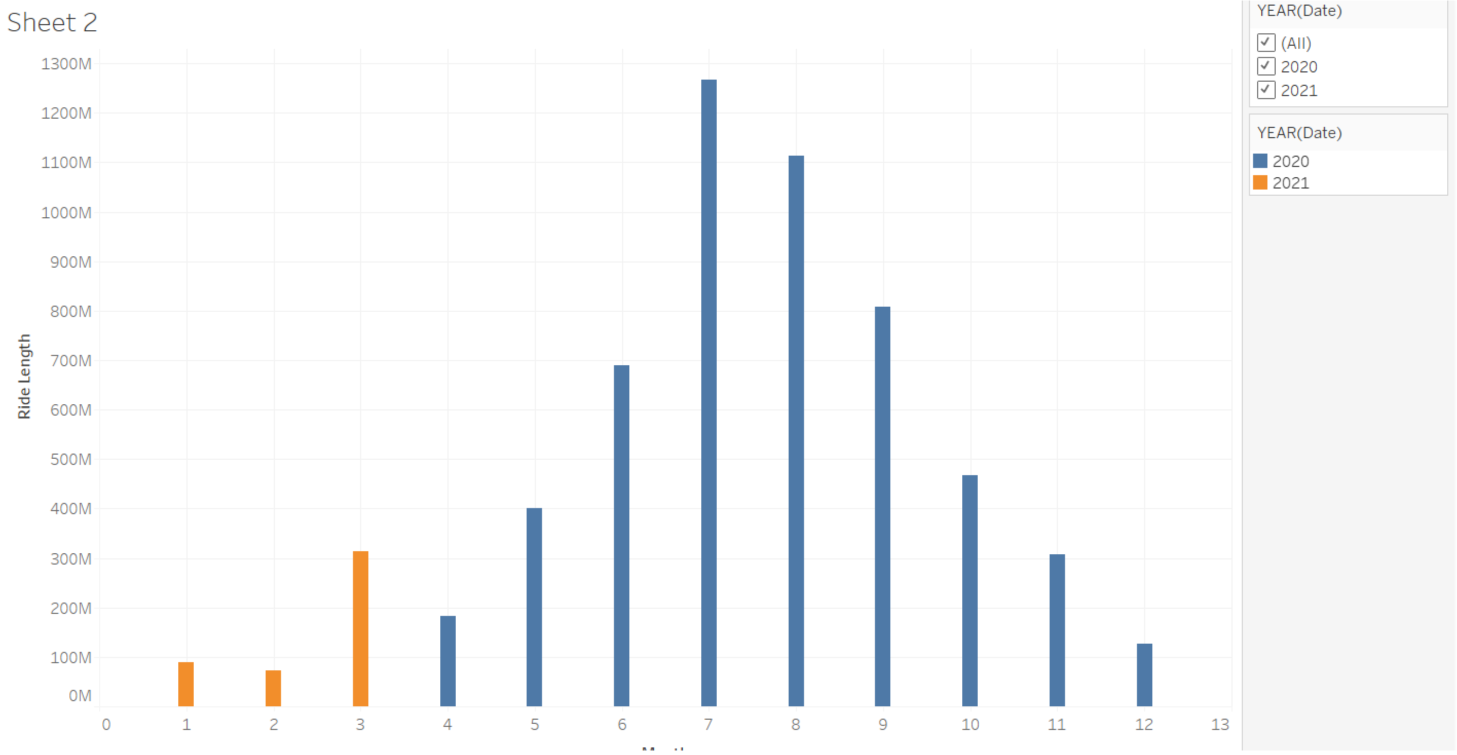


Type x Days	Weekdays	Weekends	Total
Casual	829791	**597330**	1427121
Member	**1463656**	588419	2052075
Total	2293447	1185749	3479196

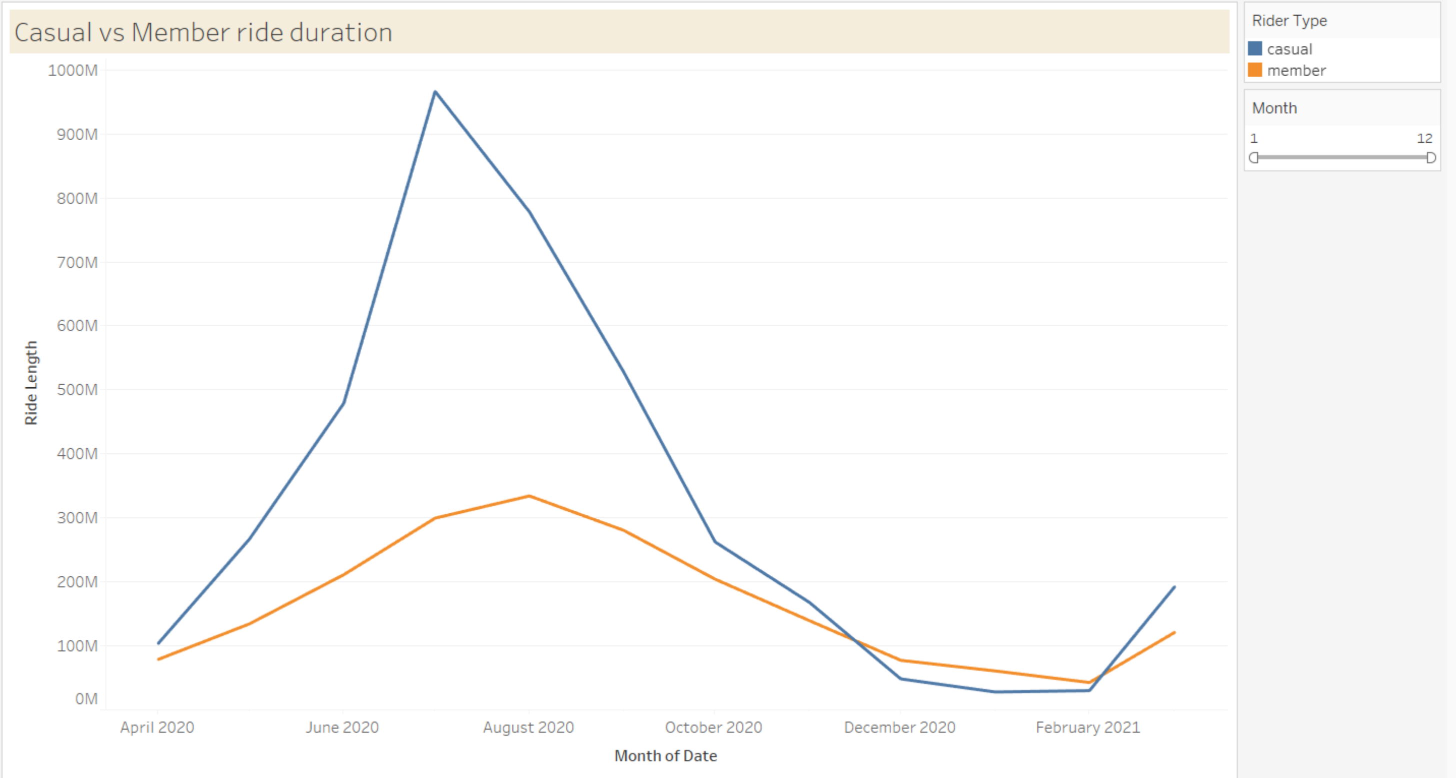
####In next phase we will share our findings through visuals by using both Tableau and Excel visualisation.

#Data Visualisation using Tableau and Excel

Below is the bar graph for Month vs Ride duration length. It can be seen that July is having the highest and Jan the lowest.

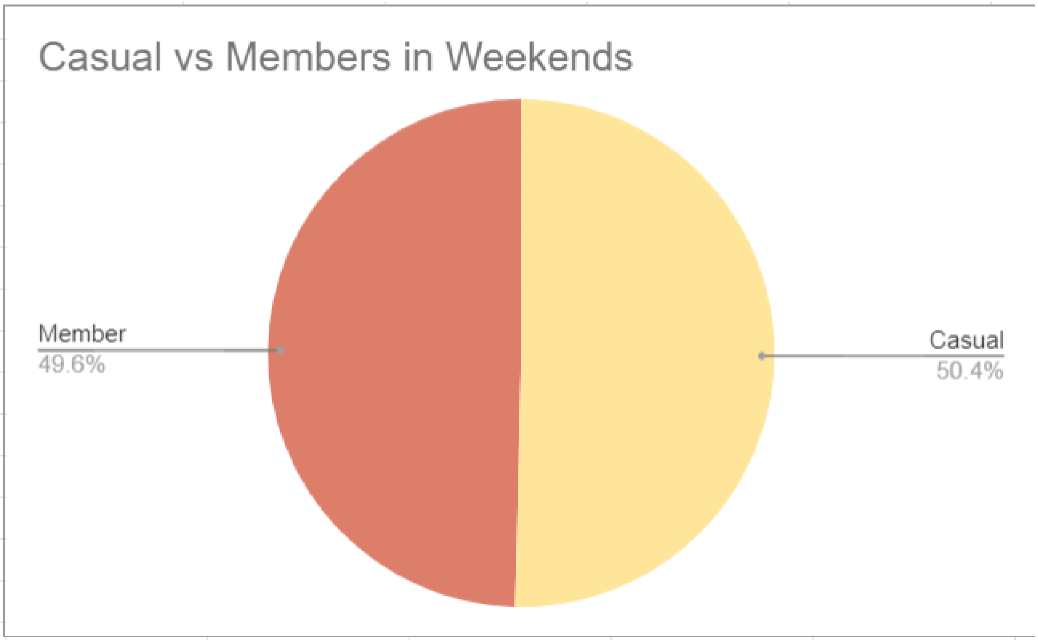


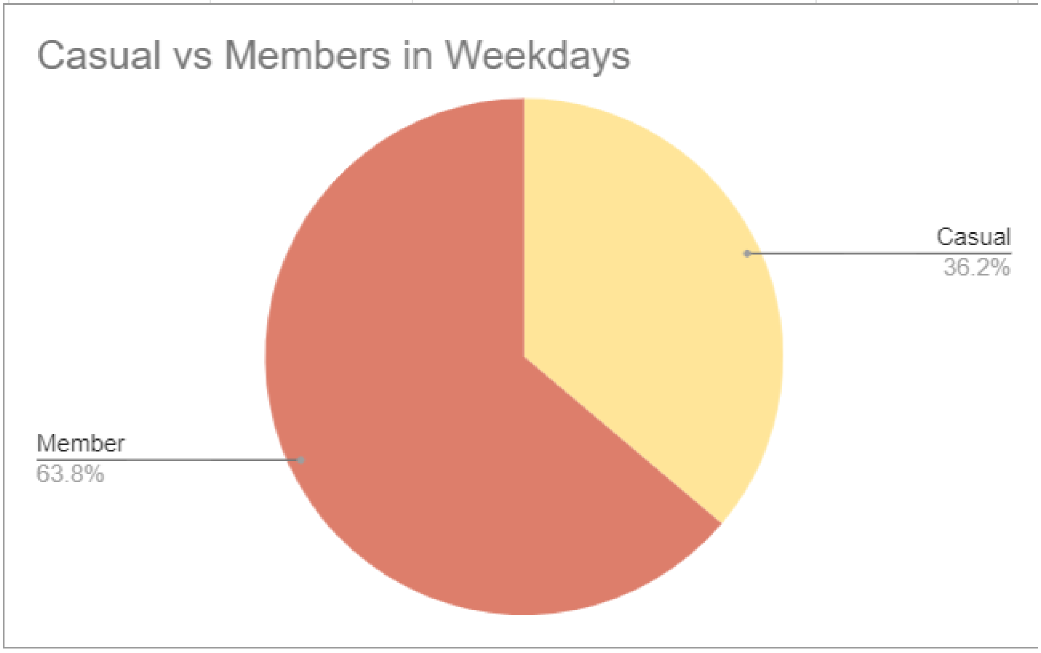
Below is the line graph for both Casual and Member rider_type. It can be seen that the ride duration increases in the mid of the year and it is lowest at the year end and year start.



It also suggest that weather is influencing the ride duration trend. In winter it's the lowest and in spring is the highest.

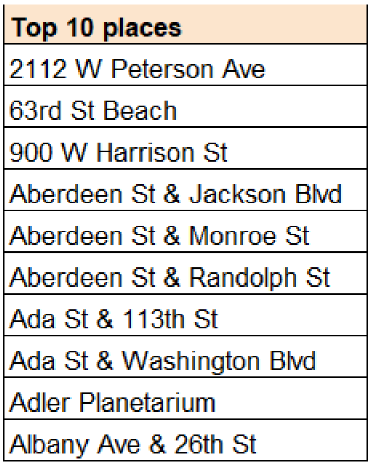
Below is the pie Chart for casual vs member in weekdays and weekends in terms of number of rides.





###The above visuals suggests that in weekdays more number of rides are coming from Members in weekdays and in weekends there is almost equal number of rides for both.

#Acting Phase 1) We can see spikes in weekends for Casual riders so we can bring different benefit schemes on weekends so that they we be attracted towards our service and will buy the membership. 2) We should also try to implement different schemes to attract the casual riders in our pick season(spring i.e.Jun,Jul,Aug). 3) Below are the top 10 stations used by casual riders. We can target this stations in other to market ourselves and attract them.



Top 10 places
2112 W Peterson Ave
63rd St Beach
900 W Harrison St
Aberdeen St & Jackson Blvd
Aberdeen St & Monroe St
Aberdeen St & Randolph St
Ada St & 113th St
Ada St & Washington Blvd
Adler Planetarium
Albany Ave & 26th St

####Thank you!

#Thank you for going through my whole documentation. I enjoyed a lot and thanks to Google Data Analytics tutors for making our journey memorable and making us enjoy through this phase.

####by Budhiman Dang