Google Capstone (Cyclistic using R)

Olexiy Pukhov

3/1/2022

Introduction to the Case Study

In this case study, I am working for a fictional company, Cyclistic, and I am asked to answer key business questions. I am asked to follow the steps of the data analysis process - ask, prepare, process, analyze, share and act.

Specifically, I am working in a Marketing analyst team at Cyclistic, a bike-sharing company in Chicago. The director of the marketing believes the company's future success depends on maximizing the number of annual memberships. My team is interested in the different trends of how casual riders and annual riders use Cyclistic Bikes differently. They want to design a new marketing strategy to convert casual riders into annual members.

I am asked to write a report with the following:

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

Step 1 - Ask

The goal of this analysis is:

– 1. Clear statement of business task: How can digital media be targeted towards casual members to incline them to get a annual membership?

Some questions to consider as well are: |. How do annual members and casual riders use cyclistic bikes differently? ||. Why would casual riders buy Cyclistic annual memberships? |||. How can Cyclistic use digital media to influence casual riders to become members

Key Stakeholders

Primary Stakeholders

Marketing Manager - Lily Moreno; a person who wants to set marketing strategies aimed at converting casual riders to annual riders. Needs data to support that decision.

Executive Team - Detail-oriented executive team that will approve my recommended marketing program.

Seconday Stakeholders

Analytics Team - A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps drive Cyclistic's marketing strategy.

Step 2 - Prepare Data

(2. A description of all the data sources used)

Cyclistic's historical trip data is located here: https://divvy-tripdata.s3.amazonaws.com/index.html In order to answer the business question, 12 months of data from 2021 will be downloaded. This corresponds to 12 files - 202101-divvy-tripdata.csv to 202112-divvy-tripdata. Although there are more recent files than 2021/12, these 12 files were chosen in order to have the most recent snapshot of the entire year.

Step 3 - Process Data

3. Documentation of any cleaning or manipulation of data

Using the programming language R, I will process, filter, and analyze the data. R is a powerful tool for data analysis because it is flexible, reproducible, and optimized for cleaning and visualizing large data.

```
knitr::opts_chunk$set(dpi = 300)

# install packages and load libraries
if(!require(tidyverse)) install.packages("tidyverse")

## Loading required package: tidyverse

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

if(!require(rio)) install.packages("rio")

## Loading required package: rio

if(!require(skimr))install.packages("skimr")

## Loading required package: skimr

if(!require(rmarkdown))install.packages("rmarkdown")

## Loading required package: rmarkdown
```

```
if(!require(doParallel))install.packages("doParallel")
```

```
## Loading required package: doParallel
```

```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
if(!require(lubridate))install.packages("lubridate")
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
if(!require(patchwork))install.packages("patchwork")
```

```
## Loading required package: patchwork
```

```
library(doParallel)
library(skimr)
library(lubridate)
library(rmarkdown)
library(tidyverse)
library(rio)
library(patchwork)
#rio is a newer version of read_csv, and allows you
#to import all sorts of file extensions. It also
#correctly imports the date, instead of characters
#as read_csv.

# do parallel computing in order to make model and
# plot calculation faster.
c1 <- makePSOCKcluster(5)
registerDoParallel(c1)
options(scipen=999)
```

We will start with importing the data.

```
##What I wrote below needs to feed into something,
#so I imported the first entry.
```

```
data <- import("202101-divvy-tripdata.csv")

for (i in 2:12) {
  if (i >= 10) {
    path = paste("2021", as.character(i), "-divvy-tripdata.csv", sep="")
    newdata <- import(path) }
  else {
    path = paste("20210", as.character(i), "-divvy-tripdata.csv", sep="")
    newdata <- import(path) }
  data = rbind(data,newdata)}
```

Now, let us look through the data.

```
head(data)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 E19E6F1B8D4C42ED electric_bike 2021-01-23 16:14:19 2021-01-23 16:24:44
## 2 DC88F20C2C55F27F electric_bike 2021-01-27 18:43:08 2021-01-27 18:47:12
## 3 EC45C94683FE3F27 electric_bike 2021-01-21 22:35:54 2021-01-21 22:37:14
## 4 4FA453A75AE377DB electric_bike 2021-01-07 13:31:13 2021-01-07 13:42:55
## 5 BE5E8EB4E7263A0B electric_bike 2021-01-23 02:24:02 2021-01-23 02:24:45
## 6 5D8969F88C773979 electric_bike 2021-01-09 14:24:07 2021-01-09 15:17:54
##           start_station_name start_station_id end_station_name
end_station_id
## 1 California Ave & Cortez St            17660
## 2 California Ave & Cortez St            17660
## 3 California Ave & Cortez St            17660
## 4 California Ave & Cortez St            17660
## 5 California Ave & Cortez St            17660
## 6 California Ave & Cortez St            17660
##   start_lat start_lng end_lat end_lng member_casual
## 1  41.90034 -87.69674   41.89  -87.72        member
## 2  41.90033 -87.69671   41.90  -87.69        member
## 3  41.90031 -87.69664   41.90  -87.70        member
## 4  41.90040 -87.69666   41.92  -87.69        member
## 5  41.90033 -87.69670   41.90  -87.70        casual
## 6  41.90041 -87.69676   41.94  -87.71        casual
```

```
str(data)
```

```
## 'data.frame':    5595063 obs. of  13 variables:
##  $ ride_id           : chr  "E19E6F1B8D4C42ED" "DC88F20C2C55F27F"
"EC45C94683FE3F27" "4FA453A75AE377DB" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct, format: "2021-01-23 16:14:19" "2021-01-27
18:43:08" ...
##  $ ended_at          : POSIXct, format: "2021-01-23 16:24:44" "2021-01-27
18:47:12" ...
##  $ start_station_name: chr  "California Ave & Cortez St" "California Ave &
Cortez St" "California Ave & Cortez St" "California Ave & Cortez St" ...
##  $ start_station_id  : chr  "17660" "17660" "17660" "17660" ...
```

```
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```
glimpse(data)
```

```
## Rows: 5,595,063
## Columns: 13
## $ ride_id            <chr> "E19E6F1B8D4C42ED", "DC88F20C2C55F27F",
"EC45C94683~
## $ rideable_type      <chr> "electric_bike", "electric_bike",
"electric_bike", ~
## $ started_at         <dttm> 2021-01-23 16:14:19, 2021-01-27 18:43:08,
2021-01-~
## $ ended_at           <dttm> 2021-01-23 16:24:44, 2021-01-27 18:47:12,
2021-01-~
## $ start_station_name <chr> "California Ave & Cortez St", "California Ave &
Cor~
## $ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660",
"17660~
## $ end_station_name   <chr> "", "", "", "", "", "", "", "", "", "Wood St &
Augu~
## $ end_station_id     <chr> "", "", "", "", "", "", "", "", "", "657",
"13258",~
## $ start_lat          <dbl> 41.90034, 41.90033, 41.90031, 41.90040,
41.90033, 4~
## $ start_lng          <dbl> -87.69674, -87.69671, -87.69664, -87.69666, -
87.696~
## $ end_lat            <dbl> 41.89000, 41.90000, 41.90000, 41.92000,
41.90000, 4~
## $ end_lng            <dbl> -87.72000, -87.69000, -87.70000, -87.69000, -
87.700~
## $ member_casual      <chr> "member", "member", "member", "member",
"casual", "~
```

```
summary(data)
```

```
##    ride_id          rideable_type        started_at
##  Length:5595063     Length:5595063     Min.   :2021-01-01 00:02:05
##  Class :character   Class :character   1st Qu.:2021-06-06 23:52:40
##  Mode  :character   Mode  :character   Median :2021-08-01 01:52:11
##                                        Mean   :2021-07-29 07:41:02
##                                        3rd Qu.:2021-09-24 16:36:16
##                                        Max.   :2021-12-31 23:59:48
##
##     ended_at                    start_station_name start_station_id
##  Min.   :2021-01-01 00:08:39   Length:5595063     Length:5595063
##  1st Qu.:2021-06-07 00:44:21   Class :character   Class :character
```

```
##  Median :2021-08-01 02:21:55   Mode  :character   Mode  :character
##  Mean   :2021-07-29 08:02:58
##  3rd Qu.:2021-09-24 16:54:05
##  Max.   :2022-01-03 17:32:18
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:5595063     Length:5595063     Min.   :41.64   Min.   :-87.84
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64
##                                        Mean   :41.90   Mean   :-87.65
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :42.07   Max.   :-87.52
##
##     end_lat         end_lng       member_casual
##  Min.   :41.39   Min.   :-88.97   Length:5595063
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character
##  Median :41.90   Median :-87.64   Mode  :character
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.17   Max.   :-87.49
##  NA's   :4771    NA's   :4771
```

```
skim(data)
```

Data summary

Name	data
Number of rows	5595063
Number of columns	13

_____

Column type frequency:	
character	7
numeric	4
POSIXct	2

_____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1	16	16	0	5595063	0
rideable_type	0	1	11	13	0	3	0
start_station_name	0	1	0	53	690809	848	0
start_station_id	0	1	0	36	69080	835	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
end_station_name	0	1	0	53	739176	845	0
end_station_id	0	1	0	36	739170	833	0
member_casual	0	1	6	6	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
start_lat	0	1	41.90	0.05	41.64	41.88	41.90	41.93	42.07	▁▁▆▇▁
start_lng	0	1	-87.65	0.03	-87.84	-87.66	-87.64	-87.63	-87.52	▁▁▆▇▁
end_lat	4771	1	41.90	0.05	41.39	41.88	41.90	41.93	42.17	▁▁▁▁▇▁
end_lng	4771	1	-87.65	0.03	-88.97	-87.66	-87.64	-87.63	-87.49	▁▁▁▁▇

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2021-01-01 00:02:05	2021-12-31 23:59:48	2021-08-01 01:52:11	4677998
ended_at	0	1	2021-01-01 00:08:39	2022-01-03 17:32:18	2021-08-01 02:21:55	4671372

This is big data, with 559k observations over 13 columns. Now let us process the data. There might be some NAs, so lets remove the columns with those and remove duplicate entries. Let us remove rows with blank information, new columns with data on month, day, hour and route and remove the information from the imported data that we don't need to make our processing faster.

```
data = na.omit(data)
data = distinct(data)

data = data %>%
  filter(start_station_id != "" & end_station_name != "" &
           start_station_name != "" & end_station_id != "" ) %>%
```

```
  mutate(duration = round(difftime(ended_at, started_at, units = "mins")))
%>%
  mutate(month = month(started_at, label = TRUE)) %>%
  mutate(wday = wday(started_at, label = TRUE)) %>%
  mutate(hour = hour(started_at)) %>%
  mutate(route = paste(start_station_name, end_station_name, sep = " to "))
%>%
  select(-c(start_station_name, end_station_name, ride_id,
            start_station_id, end_station_id, start_lat,
            start_lng, end_lat, end_lng))
```

Some of the information in the data are strings, and can't be grouped together to be used further in the analysis. Let us make everything into factors so this is possible, remove entries with a negative trip duration, and sort everything in descending order by duration.

```
data <- as.data.frame(unclass(data), stringsAsFactors = TRUE)
data = data %>%
  filter(duration > 0) %>%
  arrange(-duration)
```

Let us group everything by month, member_casual (if they are members or casual riders), rideablee_type (there are 3 types of bikes - classic, docked and electric bikes) and look at some stats for the groups.

```
data %>%
  group_by(member_casual, wday) %>%
  summarize(mean = round(mean(duration)), med = median(duration),
            count = n())

## `summarise()` has grouped output by 'member_casual'. You can override
using the `.groups` argument.

## # A tibble: 14 x 5
## # Groups:   member_casual [2]
##    member_casual wday  mean    med      count
##    <fct>         <ord> <drtn>  <drtn>   <int>
##  1 casual        Sun   38 mins 20 mins 401446
##  2 casual        Mon   33 mins 17 mins 227583
##  3 casual        Tue   29 mins 15 mins 213663
##  4 casual        Wed   28 mins 14 mins 216898
##  5 casual        Thu   28 mins 14 mins 222924
##  6 casual        Fri   31 mins 16 mins 288407
##  7 casual        Sat   35 mins 19 mins 465725
##  8 member        Sun   15 mins 11 mins 307771
##  9 member        Mon   13 mins  9 mins 342941
## 10 member        Tue   13 mins  9 mins 384495
## 11 member        Wed   13 mins  9 mins 393914
## 12 member        Thu   12 mins  9 mins 369888
## 13 member        Fri   13 mins 10 mins 362108
## 14 member        Sat   15 mins 11 mins 353215
```

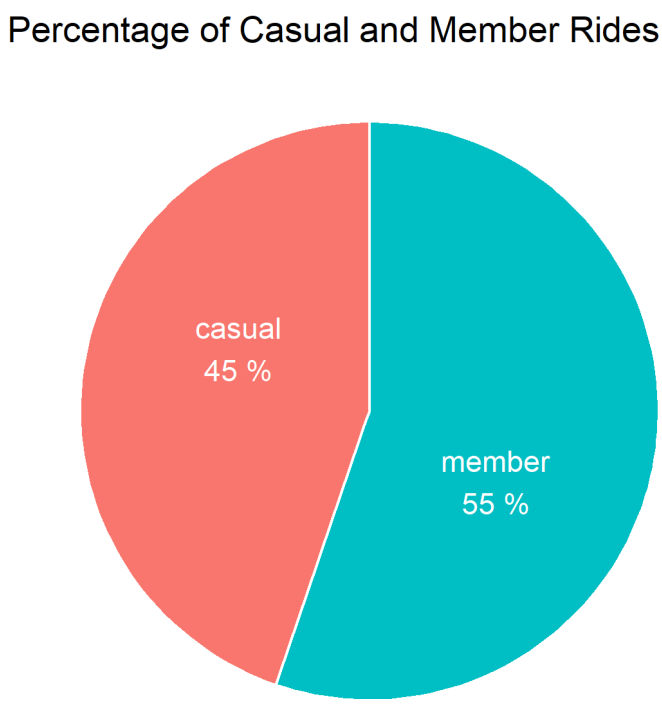
Casual members seem to ride about ~2x more than members, with the most on weekends. Members seem to ride ~15min, while casuals ride around ~30min.

Step 4 & 5 - Analyze and Visualize Data

These two steps will be done at the same time.

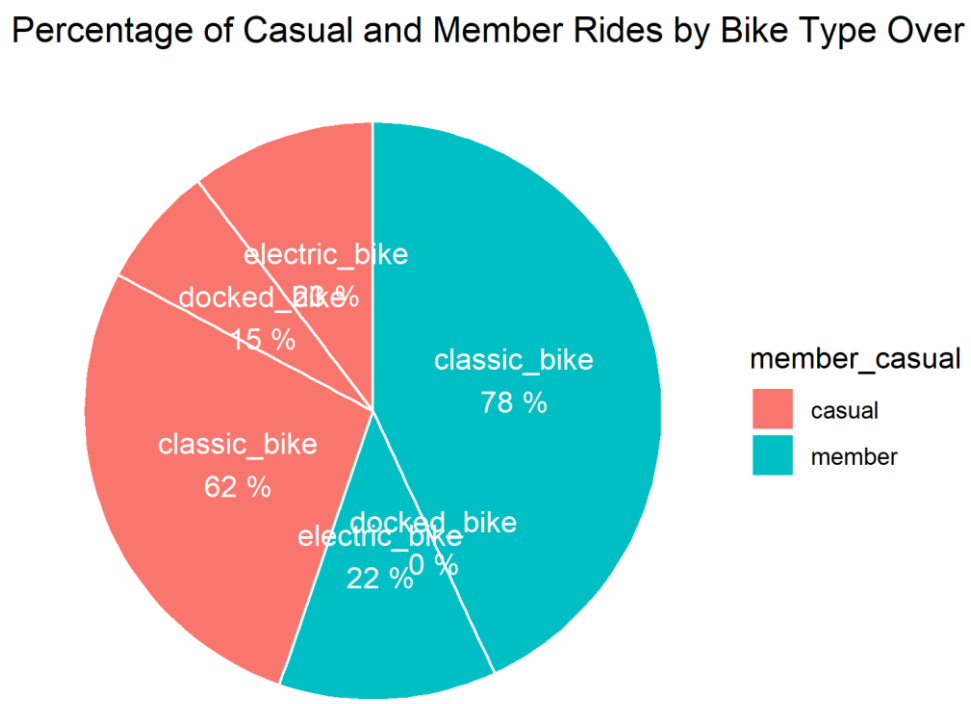
Let us start making some visuals for the data.

```
#Percentage of casual and member rides over the last year. Members
#have ridden more times than casuals. Lets break this down further.
#1. Members (55%) has ridden more rides than casuals (45%).

data %>%
  group_by(member_casual)%>%
  summarize(count = n()) %>%
  mutate(percent = round(count*100/sum(count),0))  %>%
  ggplot(aes(x = "", y = count, fill = member_casual)) +
  geom_bar(width = 1, stat = "identity", color = "white", show.legend =
FALSE) +
  coord_polar("y", start = 0) +
  geom_text(aes(label =
                  paste(member_casual,
                        paste(percent, "%"),
                        sep = "\n")),
            position = position_stack(vjust = 0.6),
            color = "white") +
  labs(title = "Percentage of Casual and Member Rides Over the Last Year")+
  theme_void()
```



. It is found that Members (55%) have ridden more rides than casuals (45%) Let's break this data down further by bike type.

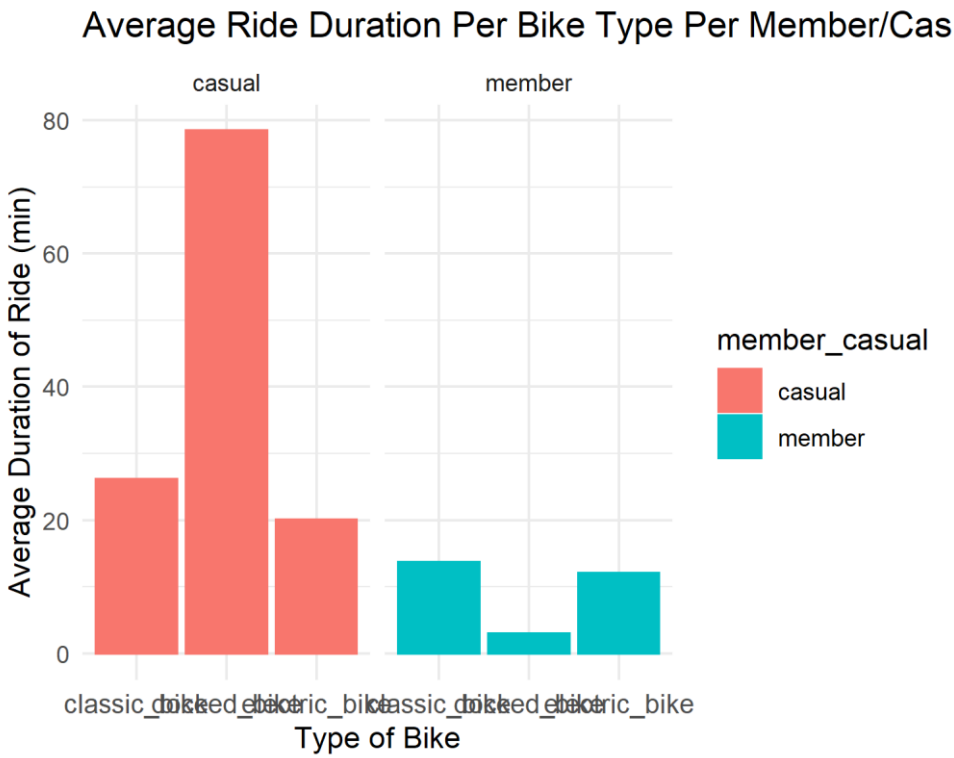
```
#Breaking down the data further by bike type. For some reason, there are no
#Docked bikes used by members.
#2. The most popular bike for members and casuals were classic bikes.

data %>%
  group_by(member_casual, rideable_type)%>%
  summarize(count = n()) %>%
  mutate(percent = round(count*100/sum(count),0))  %>%
  ggplot(aes(x = "", y = count, fill = member_casual, rideable_type)) +
  geom_bar(width = 1, stat = "identity", color = "white", show.legend = TRUE)
+
  coord_polar("y", start = 0) +
  geom_text(aes(label =
                  paste(rideable_type,
                        paste(percent, "%"),
                        sep = "\n")),
            position = position_stack(vjust = 0.5),
            color = "white") +
  labs(title = "Percentage of Casual and Member Rides by Bike Type Over the
Last Year")+
  theme_void()

## `summarise()` has grouped output by 'member_casual'. You can override
using the `.groups` argument.
```



. The most popular bike for members (78%) and casuals (62%) were classic bikes. Members did not use any docked bikes.

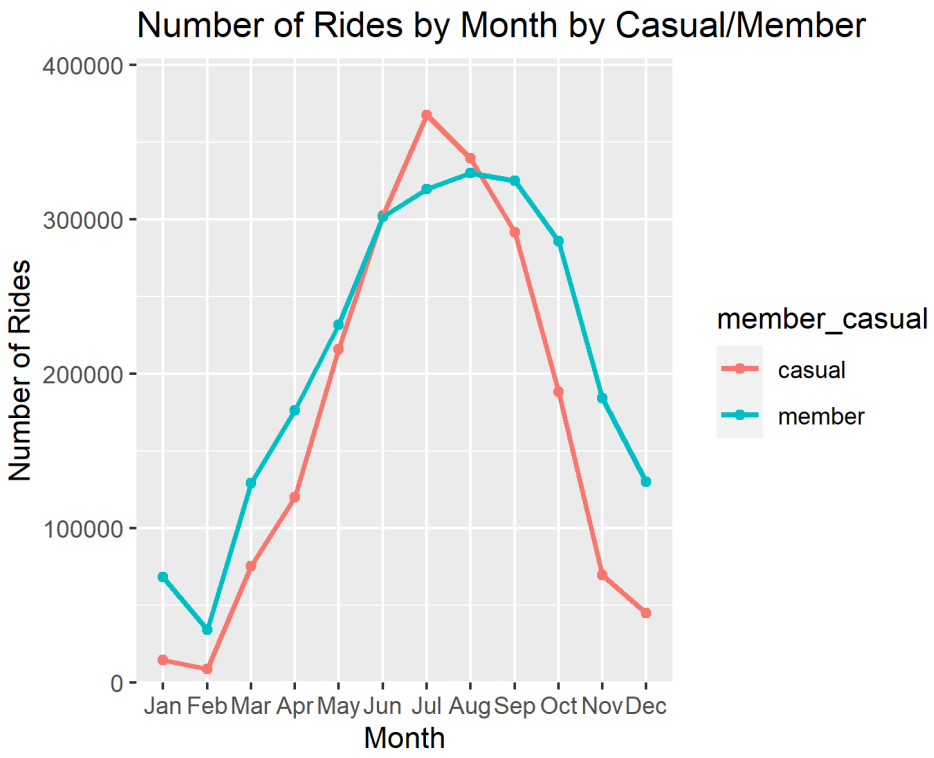
Lets investigate the duration of bike rides.

```
#Lets do the same plot but now for percentage of duration of bike rides.
#3. Although classic bikes are the most popular, docked bikes had the
#greatest duration in the casual member groups. They were barely used in the
#members group.

data %>%
  group_by(member_casual, rideable_type) %>%
  summarize(dur = mean(duration), count = n()) %>%
  ggplot(aes(x = rideable_type, y = dur,
             color = member_casual,
             fill = member_casual))+
  geom_bar(stat = "identity") +
  facet_wrap(~member_casual) +
  labs (title = "Average Ride Duration Per Bike Type Per Member/Casual",
        x = "Type of Bike", y = "Average Duration of Ride (min)") +
  theme_minimal() +
  theme(axis.text.x =
          element_text(size = 10))

## `summarise()` has grouped output by 'member_casual'. You can override
using the `.groups` argument.

## Don't know how to automatically pick scale for object of type difftime.
Defaulting to continuous.
```



. We have found another insight. Although classic bikes are the most popular, docked bikes had the greatest duration in the casual member groups. They were barely used in the members group.

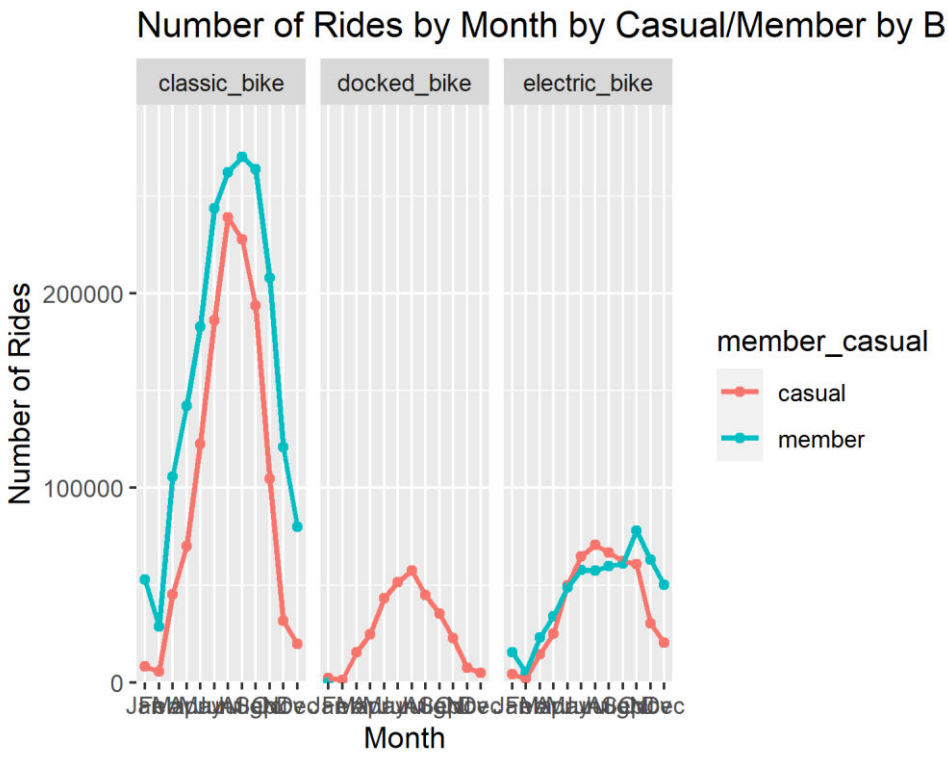
```
#Lets explore the number of rides per member/casual per month.
#4. Members rode more times than casuals using classic bikes. Members did
#not use docked bikes. In the summer (Jun to Aug), casual rides were greater
than
#member rides. This was mostly driven by classic bikes in the casuals group,
# and partly by docked bikes in the casuals group that was absent in the
members
# group.

data %>%
  group_by(month, member_casual) %>%
  summarize(count = n()) %>%
  ggplot +
  aes(x=month, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Number of Rides by Month by Casual/Member",
        x = "Month", y = "Number of Rides")

## `summarise()` has grouped output by 'month'. You can override using the
`.groups` argument.
```



. Members rode more times than casuals using classic bikes. Members did not use docked bikes. In the summer (Jun to Aug), casual rides were greater than member rides. Let us break this down further into rideable bike type for more info.

```
  data %>%
  group_by(month, member_casual,rideable_type) %>%
  summarize(count = n()) %>%
  ggplot +
  aes(x=month, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Number of Rides by Month by Casual/Member by Bike Type",
        x = "Month", y = "Number of Rides") +
  facet_wrap(~rideable_type)
```

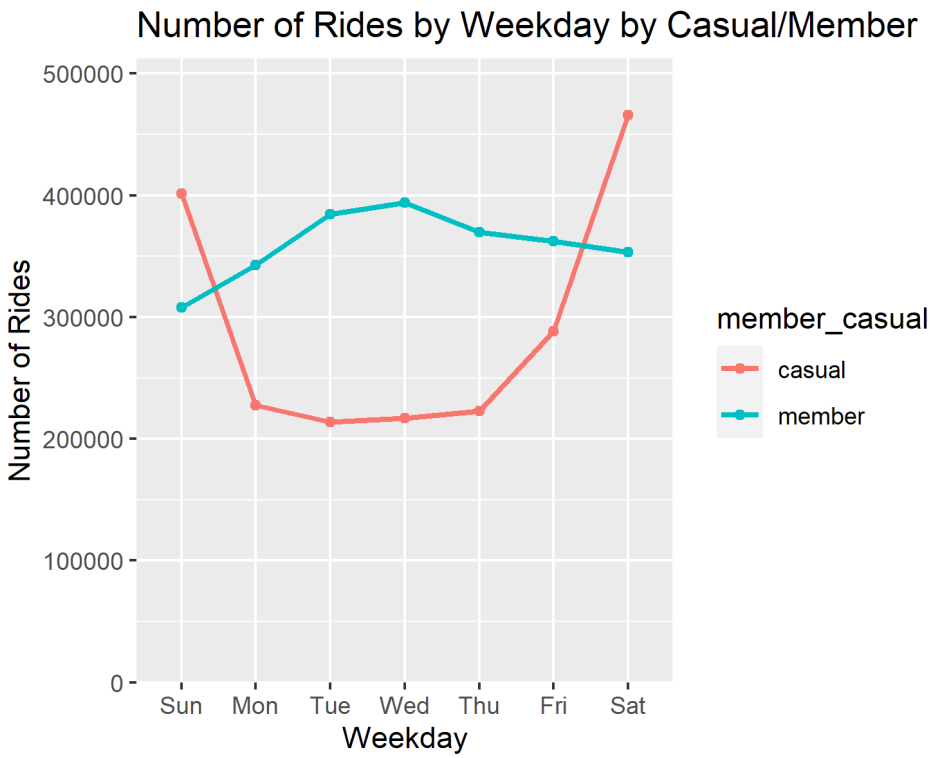
```
## `summarise()` has grouped output by 'month', 'member_casual'. You can
override using the `.groups` argument.
```



. In the analysis of #4, this was mostly driven by classic bikes in the casuals group, and partly by docked bikes in the casuals group. Docked bikes were absent in the members group.

Let us look at the number of rides per member/casual per weekday.

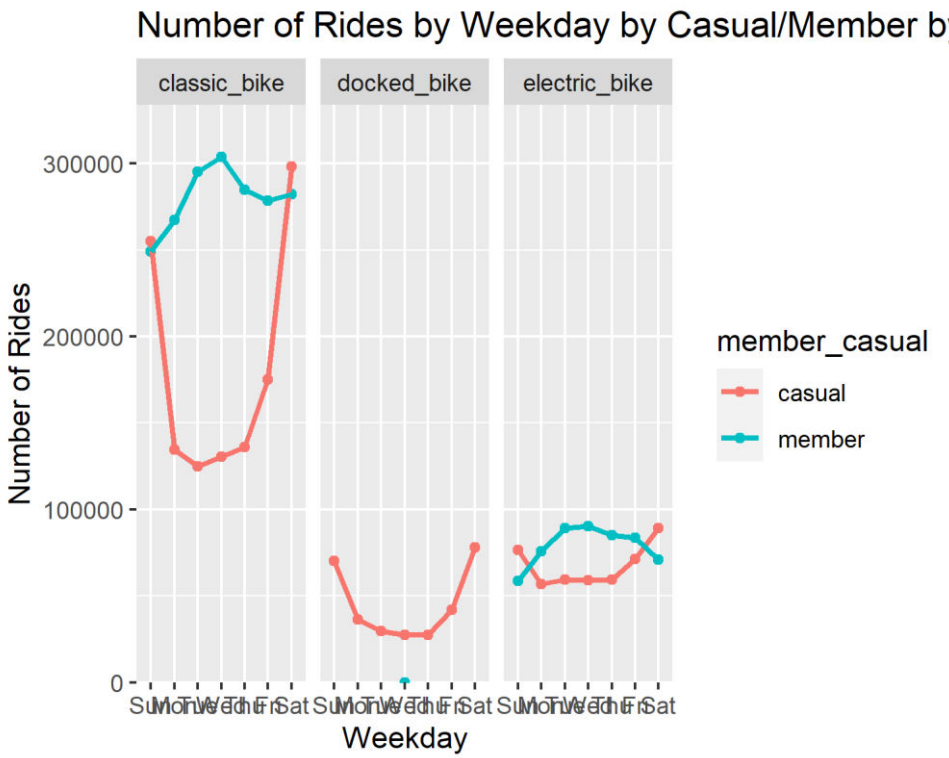
```
#Lets explore the number of rides per member/casual per weekday.
#6. Members rode the most times in the middle of the week, being fairly
consistent
# throughout the whole week. Casuals rode the most on weekends.

data %>%
  group_by(wday, member_casual) %>%
  summarize(count = n()) %>%
  ggplot +
  aes(x=wday, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Number of Rides by Weekday by Casual/Member",
        x = "Weekday", y = "Number of Rides")

## `summarise()` has grouped output by 'wday'. You can override using the
`.groups` argument.
```



Let us break this data down by bike type.

```
  data %>%
  group_by(wday, member_casual,rideable_type) %>%
  summarize(count = n()) %>%
  ggplot +
  aes(x=wday, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Number of Rides by Weekday by Casual/Member by Bike Type",
        x = "Weekday", y = "Number of Rides") +
  facet_wrap(~rideable_type)
```

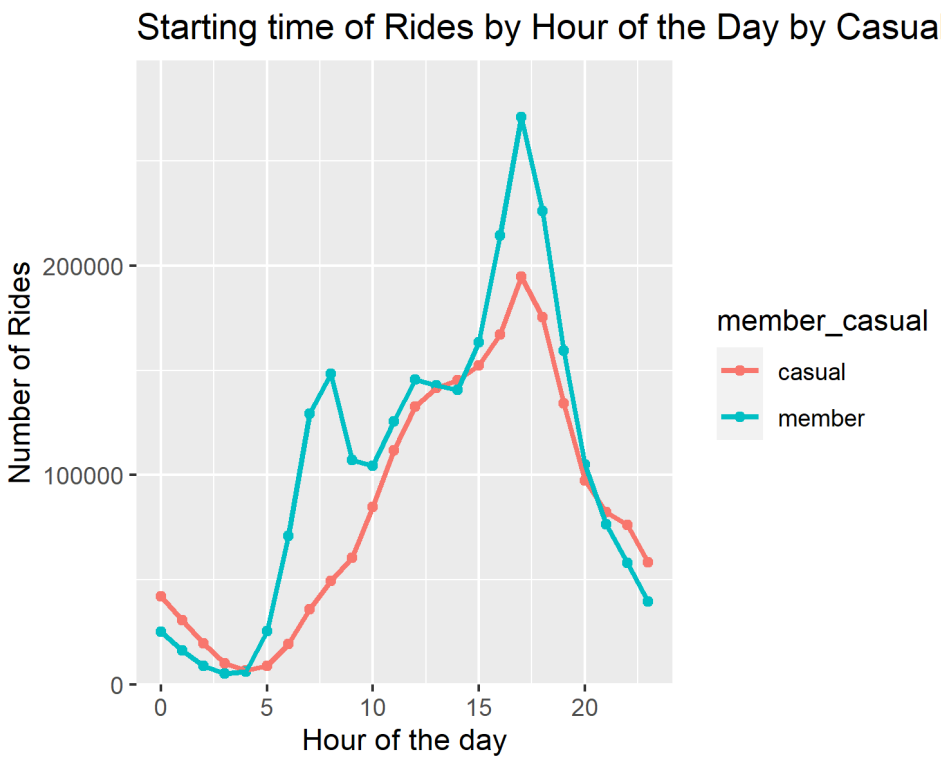
```
## `summarise()` has grouped output by 'wday', 'member_casual'. You can
override using the `.groups` argument.
```



. Members rode the most times in the middle of the week, being fairly consistent throughout the whole week. Casuals rode the most on weekends.

Let us look at the number of rides per member/casual per starting time hour of the day.

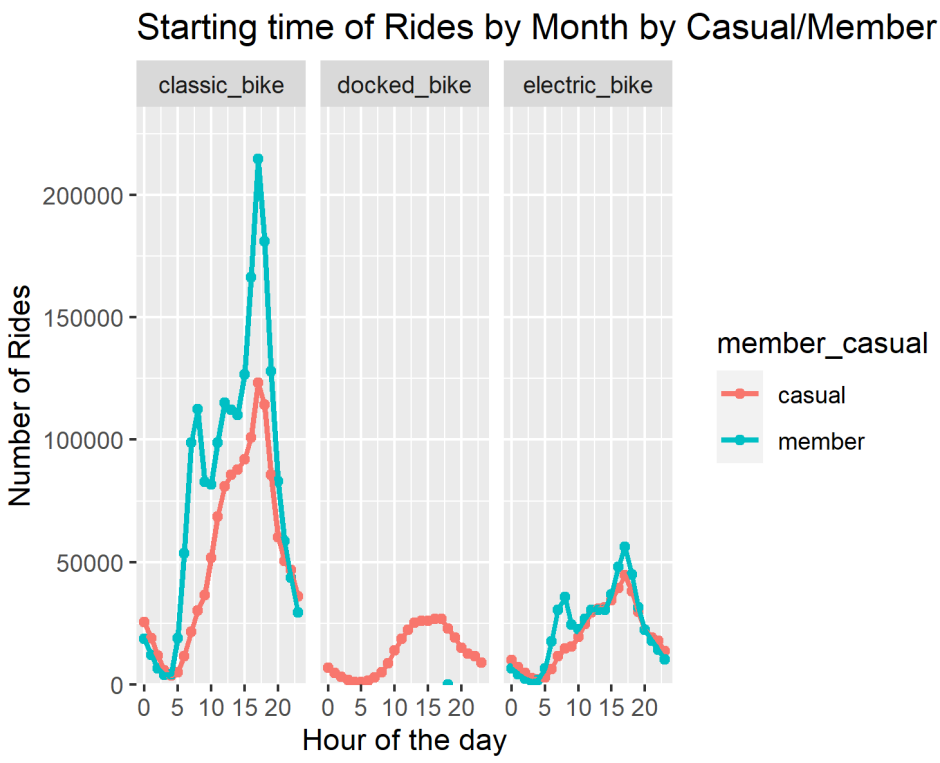
```
#Lets explore the number of rides per member/casual per hour of the day.
#7. Casuals and members took the most rides 3-8pm.

data %>%
  group_by(hour, member_casual) %>%
  summarize(count = n()) %>%
  ggplot +
  aes(x=hour, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Starting time of Rides by Hour of the Day by Casual/Member",
        x = "Hour of the day", y = "Number of Rides")

## `summarise()` has grouped output by 'hour'. You can override using the
`.groups` argument.
```



Let us break this data down by bike type.

```
  data %>%
  group_by(hour, member_casual,rideable_type) %>%
  summarize(count = n()) %>%
  ggplot +
  aes(x=hour, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Starting time of Rides by Month by Casual/Member by Bike
Type",
        x = "Hour of the day", y = "Number of Rides") +
  facet_wrap(~rideable_type)
```

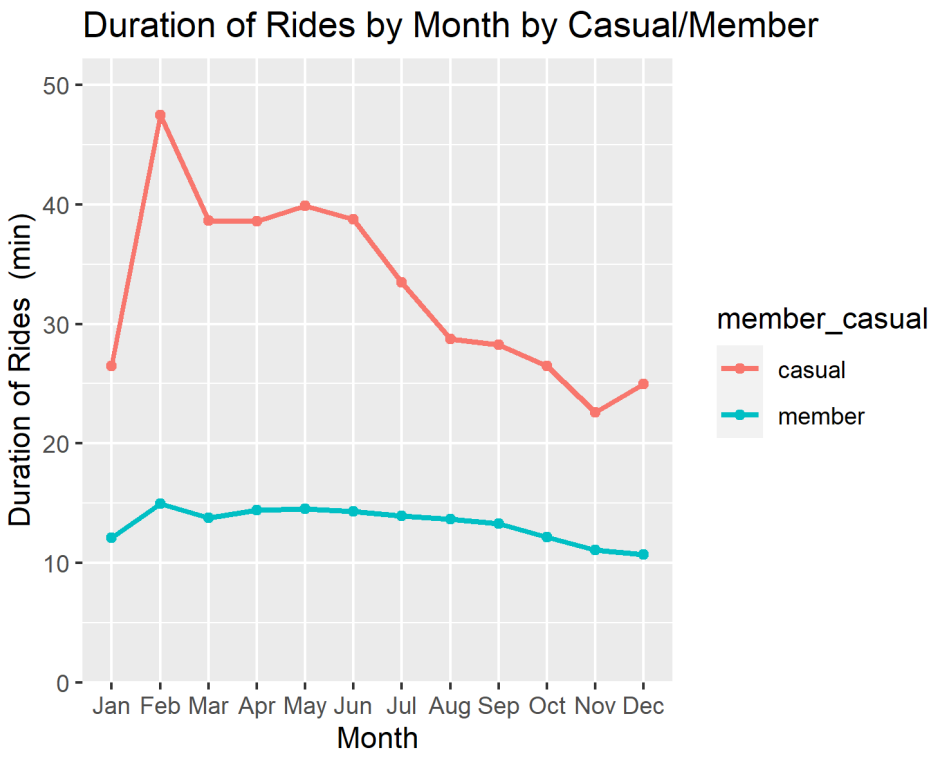
```
## `summarise()` has grouped output by 'hour', 'member_casual'. You can
override using the `.groups` argument.
```



. Casuals and members took the most rides 3-8pm.

Lets look at the duration of the rides per member/casual per month.

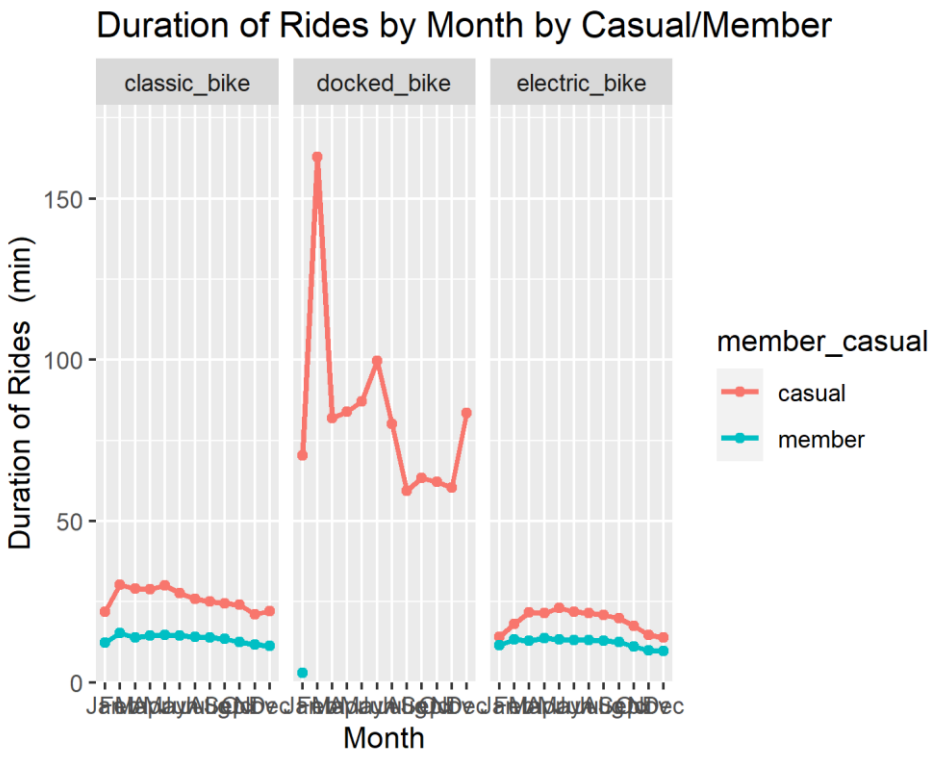
```
# Same thing, but with duration

#Lets explore the duration of rides per member/casual per month.
#8. Casual rides on average were about 2x longer (~30min) than members
throughout the year (~15min).
# The longest rides for casuals were driven by docked bikes and these long
bike rides
# were undertaken with the longest duration during February and July.

data %>%
  group_by(month, member_casual) %>%
  summarize(count = mean(duration)) %>%
  ggplot +
  aes(x=month, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Duration of Rides by Month by Casual/Member",
        x = "Month", y = "Duration of Rides  (min)")

## `summarise()` has grouped output by 'month'. You can override using the
`.groups` argument.
```



Let's break this data down by bike type.

```
  data %>%
  group_by(month, member_casual,rideable_type) %>%
  summarize(count = mean(duration)) %>%
  ggplot +
  aes(x=month, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Duration of Rides by Month by Casual/Member",
        x = "Month", y = "Duration of Rides  (min)") +
  facet_wrap(~rideable_type)
```

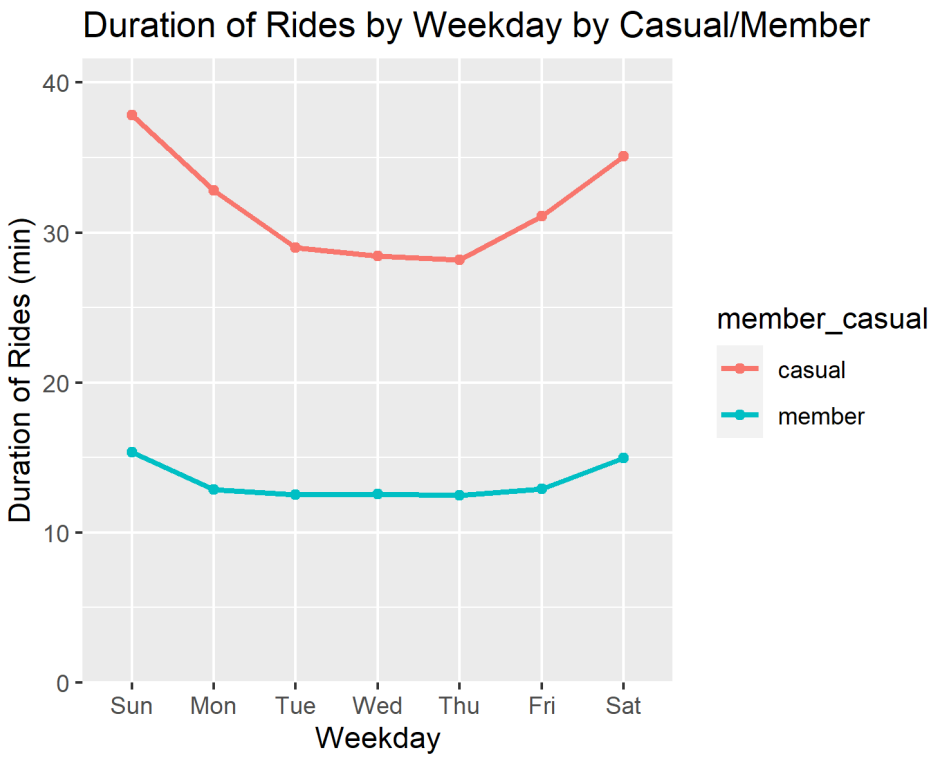
```
## `summarise()` has grouped output by 'month', 'member_casual'. You can
override using the `.groups` argument.
```



. Casual rides on average were about 2x longer (~30min) than members throughout the year (~15min). The longest rides for casuals were driven by docked bikes and these long bike rides were undertaken with the longest duration during February and July. The average duration for docked bike casual rides was ~80min, with a spike in duration to ~155min in February.

Lets explore the duration of rides per member/casual per weekday.

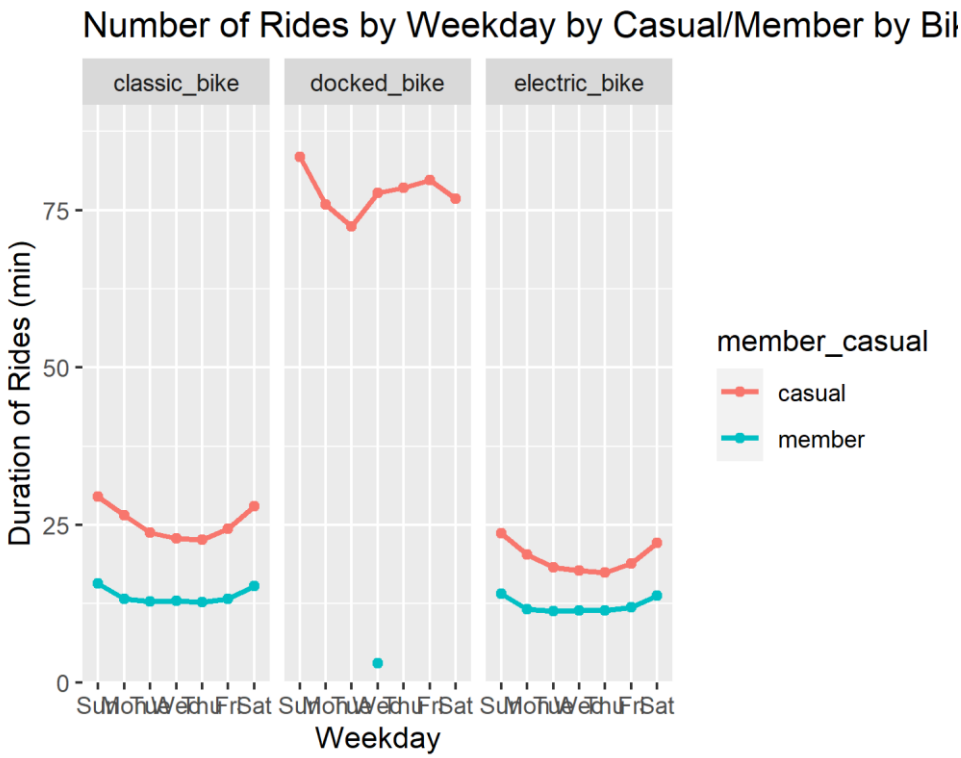
```
#Lets explore the duration of rides per member/casual per weekday.
#9. Members were consistent in their duration across the week, casuals had
the
#greatest duration on weekends. Duration of Casual Rides increased to ~35min
on weekends from ~30 min.

data %>%
  group_by(wday, member_casual) %>%
  summarize(count = mean(duration)) %>%
  ggplot +
  aes(x=wday, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Duration of Rides by Weekday by Casual/Member",
        x = "Weekday", y = "Duration of Rides (min)")

## `summarise()` has grouped output by 'wday'. You can override using the
`.groups` argument.
```



Let's break this data down by rideable type.

```
  data %>%
  group_by(wday, member_casual,rideable_type) %>%
  summarize(count = mean(duration)) %>%
  ggplot +
  aes(x=wday, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Number of Rides by Weekday by Casual/Member by Bike Type",
        x = "Weekday", y = "Duration of Rides (min)") +
  facet_wrap(~rideable_type)
```

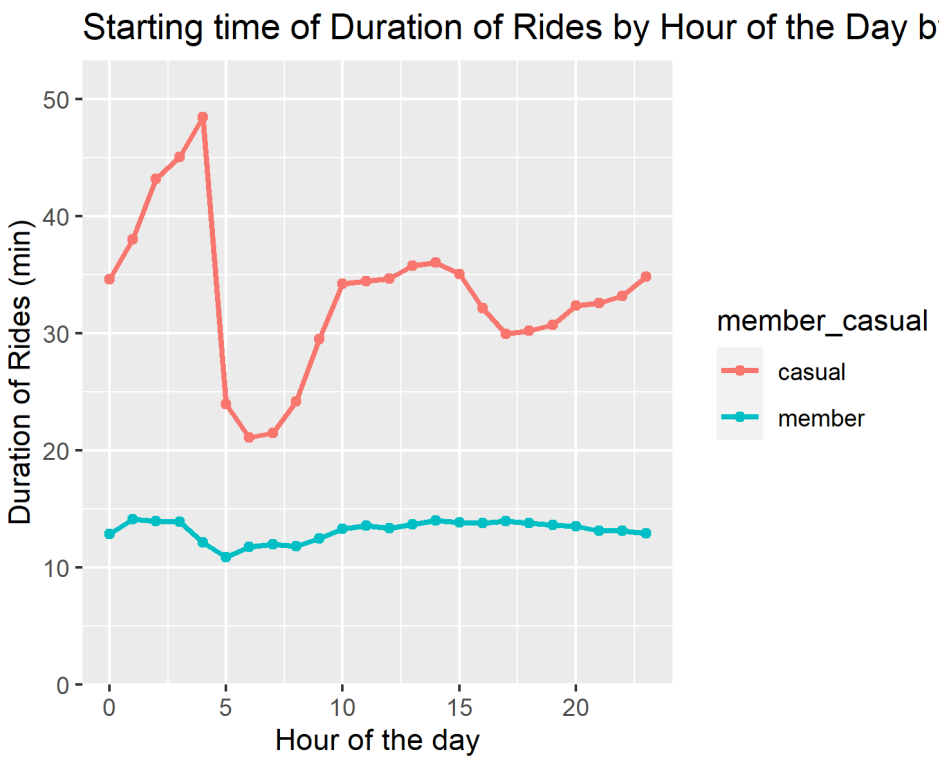
```
## `summarise()` has grouped output by 'wday', 'member_casual'. You can
override using the `.groups` argument.
```



. Members were consistent in their duration across the week, casuals had the greatest duration on weekends. Duration of Casual Rides increased to ~35min on weekends from ~30 min. Docked bike casual riders had an average of ~80min duration ride.

Lets explore the duration of rides per member/casual per hour of the day.

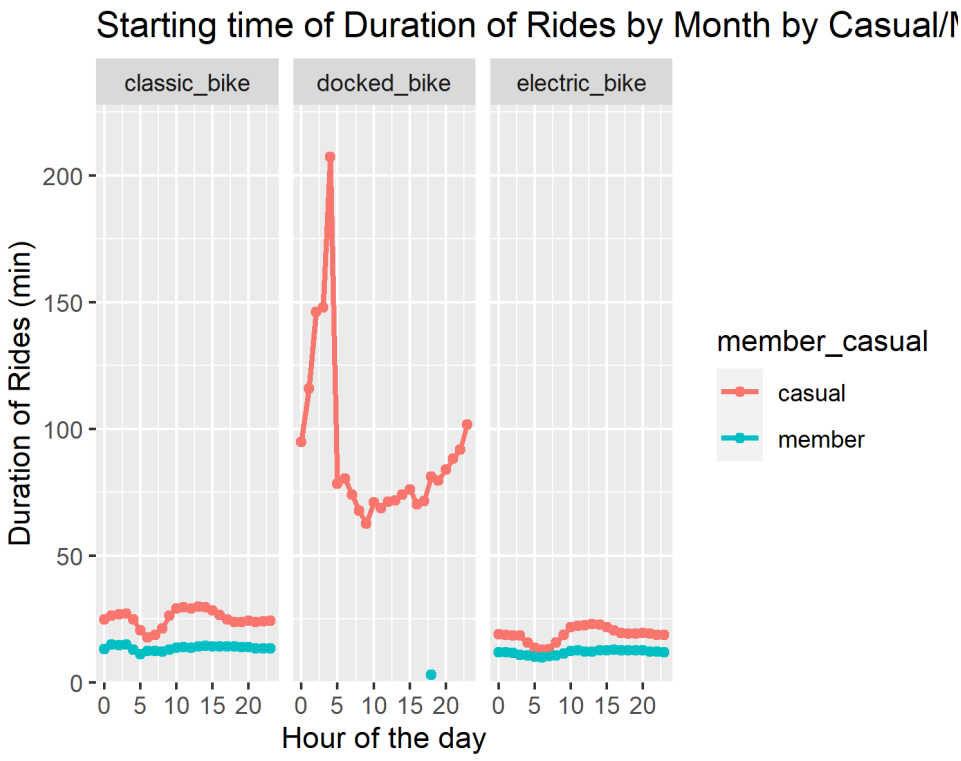
```
#Lets explore the duration of rides per member/casual per hour of the day.
#10. Casuals took the longest rides starting at 2-4am, being mostly driven by
# docked bike usage. Members were consistent with bike ride duration
throughout
# all hours.

data %>%
  group_by(hour, member_casual) %>%
  summarize(count = mean(duration)) %>%
  ggplot +
  aes(x=hour, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Starting time of Duration of Rides by Hour of the Day by
Casual/Member",
        x = "Hour of the day", y = "Duration of Rides (min)")

## `summarise()` has grouped output by 'hour'. You can override using the
`.groups` argument.
```



Let's break this down by bike type.

```
  data %>%
  group_by(hour, member_casual,rideable_type) %>%
  summarize(count = mean(duration)) %>%
  ggplot +
  aes(x=hour, y=count, color = member_casual, group = member_casual) +
  geom_point() + geom_line(size = 1) +
  scale_y_continuous(limits = c(0, NA),
                     expand = expansion(mult = c(0, 0.1))) +
  labs (title = "Starting time of Duration of Rides by Month by Casual/Member
by Bike Type",
        x = "Hour of the day", y = "Duration of Rides (min)") +
  facet_wrap(~rideable_type)
```

```
## `summarise()` has grouped output by 'hour', 'member_casual'. You can
override using the `.groups` argument.
```



. Casuals took the longest rides starting at 2-4am, being mostly driven by docked bike usage. Members were consistent with bike ride duration throughout all hours.

Let us now investigate the top 10 routes used my members, casuals and casuals that use docked bikes.

```
##Get the top 10 routes for members and casuals
casual_routes <- data %>%
  filter(member_casual == "casual") %>%
  group_by(route) %>%
  tally(sort = TRUE)

casual_routes = casual_routes[1:10,]

casual_droutes <- data %>%
  filter(member_casual == "casual") %>%
  filter(rideable_type == "docked_bike") %>%
  group_by(route) %>%
  tally(sort = TRUE)

casual_droutes = casual_droutes[1:10,]

member_routes <- data %>%
  filter(member_casual == "member") %>%
  group_by(route) %>%
  tally(sort = TRUE)

member_routes = member_routes[1:10,]
```

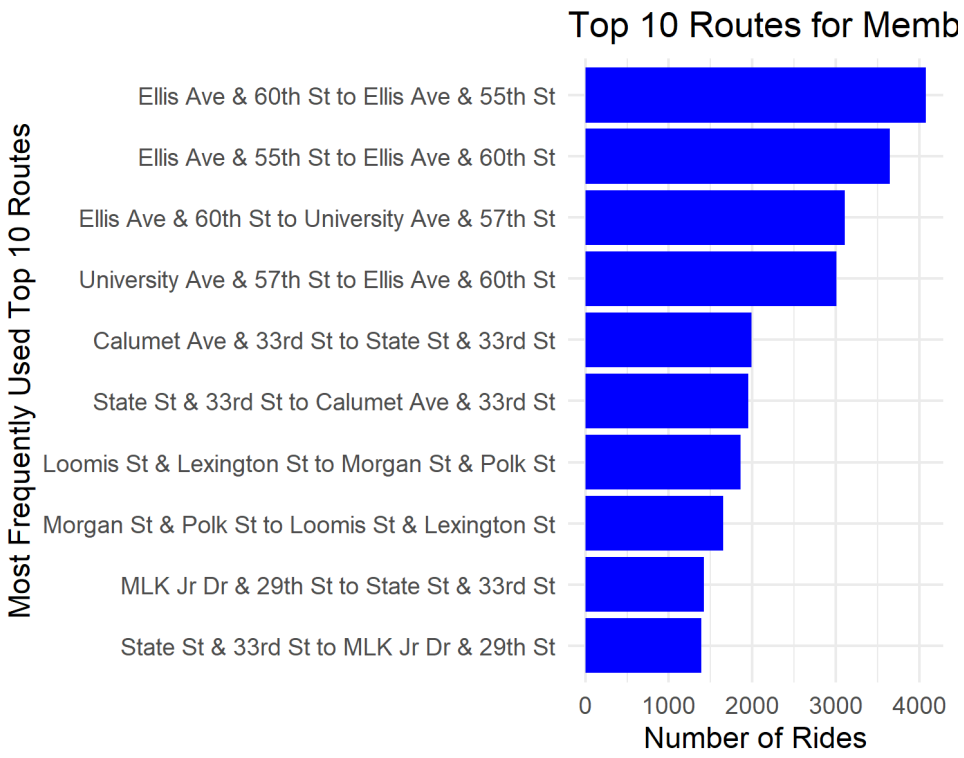
```
casual_routes %>%
  ggplot(aes(x = reorder(route, n),
             y = n)) +
  geom_bar(stat = "identity",
           fill = "blue") +
  xlab("Most Frequently Used Top 10 Routes") +
  ylab("Number of Rides") +
  coord_flip() +
  labs(title = "Top 10 Routes for Casual Riders") +
  theme_minimal()
```

11. The most popular route for casual riders was Streeter Dr & Grand Ave to Streeter Dr & Grand Ave. This seems to be a ride out to somewhere and then back to the same location.

```
casual_droutes %>%
  ggplot(aes(x = reorder(route, n),
             y = n)) +
  geom_bar(stat = "identity",
           fill = "blue") +
  xlab("Most Frequently Used Top 10 Routes") +
  ylab("Number of Rides") +
  coord_flip() +
  labs(title = "Top 10 Routes for Casual Docked Bike Riders") +
  theme_minimal()
```

. The most popular popular route for docked bike casual riders was the same as casual riders as a whole - Streeter Dr & Grand Ave to Streeter Dr & Grand Ave.

```
member_routes %>%
  ggplot(aes(x = reorder(route, n),
             y = n)) +
  geom_bar(stat = "identity",
           fill = "blue") +
  xlab("Most Frequently Used Top 10 Routes") +
  ylab("Number of Rides") +
  coord_flip() +
  labs(title = "Top 10 Routes for Member Riders") +
  theme_minimal()
```



. The most popular popular route for member riders was Ellis Ave & 60th St to Ellis Ave & 55th St, followed by Ellis and 55th St to Ellis and 60th St.

Summary of Analysis

1. It is found that Members (55%) have ridden more rides than casuals (45%)
2. The most popular bike for members (78%) and casuals (62%) were classic bikes. Members did not use any docked bikes.
3. Although classic bikes are the most popular, docked bikes had the greatest duration in the casual member groups. They were barely used in the members group.
4. Members rode more times than casuals using classic bikes. Members did not use docked bikes. In the summer (Jun to Aug), casual rides were greater than member rides.
5. In the analysis of #4, this was mostly driven by classic bikes in the casuals group, and partly by docked bikes in the casuals group. Docked bikes were absent in the members group.
6. Members rode the most times in the middle of the week, being fairly consistent throughout the whole week. Casuals rode the most on weekends.
7. Casuals and members took the most rides 3-8pm.
8. Casual rides on average were about 2x longer (~30min) than members throughout the year (~15min). The longest rides for casuals were driven by docked bikes and these long bike rides were undertaken with the longest duration during February and July. The average duration for docked bike casual rides was ~80min, with a spike in duration to ~155min in February.

9. Members were consistent in their duration across the week, casuals had the greatest duration on weekends. Duration of Casual Rides increased to ~35min on weekends from ~30 min. Docked bike casual riders had an average of ~80min duration ride.
10. Casuals took the longest rides starting at 2-4am, being mostly driven by docked bike usage. Members were consistent with bike ride duration throughout all hours.
11. The most popular route for casual riders was Streeter Dr & Grand Ave to Streeter Dr & Grand Ave. This seems to be a ride out to somewhere and then back to the same location.
12. The most popular popular route for docked bike casual riders was the same as casual riders as a whole - Streeter Dr & Grand Ave to Streeter Dr & Grand Ave.
13. The most popular popular route for member riders was Ellis Ave & 60th St to Ellis Ave & 55th St, followed by Ellis and 55th St to Ellis and 60th St.

Phase 6 - Act

Through the above analysis, I have come to the following recommendations:

1. Target digital media to riders during summer, especially during weekends between 3pm-8pm about the benefit of a annual membership that will have a discounted price depending on how much you cycle in one ride. As casual members take longer rides than members, this is an incentive.

2. Make docked bikes available to members or investigate why docked bikes are not being used by members. A significant portion of casuals use docked bikes, but not members. If we can convert some of these users into annual members, this will bring revenue.

3. Use digital media marketing to target docked bike riders especially in February and July in the morning between 2-4am around the location of Streeter Dr & Grand Ave about the price discount or advantage of such if taking a very long duration ride.

This was my first project, all feedback is appreciated. Thank you Google for the Google Data Analytics Certificate.