Case Study 1

Rithvik Shamrao Vinekar

2023-02-15

# Ask

## The stakeholders:

* **Lily Moreno:** The director of marketing and manager
* **Cyclistic marketing analytics team**
* **Cyclistic executive team**

## The question to be answered is:

* *How do annual* ***members*** *and* ***casual*** *riders use Cyclistic bikes* ***differently****?*

### These questions below are not asked, but any insight will be helpful in that direction as well

* Why would casual riders buy Cyclistic annual memberships?
* How can Cyclistic use digital media to influence casual riders to become members?

# Prepare

The data is obtained from [tripdata](https://divvy-tripdata.s3.amazonaws.com/index.html) provided for this study. We will use only the data for the year 2022, from 1st Jan 2022 to 31st Dec 2022

For this, the data in csv files is downloaded to local drive. They match the regular expression pattern “2022.\*divvy.\*.csv”:

list.files(path = "..", pattern = "2022.\*divvy.\*.csv",full.names = TRUE)

## [1] "../202201-divvy-tripdata.csv" "../202202-divvy-tripdata.csv"  
## [3] "../202203-divvy-tripdata.csv" "../202204-divvy-tripdata.csv"  
## [5] "../202205-divvy-tripdata.csv" "../202206-divvy-tripdata.csv"  
## [7] "../202207-divvy-tripdata.csv" "../202208-divvy-tripdata.csv"  
## [9] "../202209-divvy-tripdata.csv" "../202210-divvy-tripdata.csv"  
## [11] "../202211-divvy-tripdata.csv" "../202212-divvy-tripdata.csv"

# Process

To begin processing this data, we will first load the packages we need. The code below attempts to install the packages if it does not find them in the library, and loads them later

## Reading the data

The data is too large to push to github, so it has been downloaded to the top-level directory outside git repository.

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

## Loading required package: tidyverse

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(tidyverse)  
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

library(lubridate)

tripdata <-  
 list.files(path = "..", pattern = "2022.\*divvy.\*.csv",full.names = TRUE) %>%   
 map\_df(~read\_csv(.))

## Checking the data

Lets look at the data

names(tripdata)

## [1] "ride\_id" "rideable\_type" "started\_at"   
## [4] "ended\_at" "start\_station\_name" "start\_station\_id"   
## [7] "end\_station\_name" "end\_station\_id" "start\_lat"   
## [10] "start\_lng" "end\_lat" "end\_lng"   
## [13] "member\_casual"

These are column names

str(tripdata)

## spc\_tbl\_ [5,667,717 × 13] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ride\_id : chr [1:5667717] "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED419105406" ...  
## $ rideable\_type : chr [1:5667717] "electric\_bike" "electric\_bike" "classic\_bike" "classic\_bike" ...  
## $ started\_at : POSIXct[1:5667717], format: "2022-01-13 11:59:47" "2022-01-10 08:41:56" ...  
## $ ended\_at : POSIXct[1:5667717], format: "2022-01-13 12:02:44" "2022-01-10 08:46:17" ...  
## $ start\_station\_name: chr [1:5667717] "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & Fullerton Ave" "Clark St & Bryn Mawr Ave" ...  
## $ start\_station\_id : chr [1:5667717] "525" "525" "TA1306000016" "KA1504000151" ...  
## $ end\_station\_name : chr [1:5667717] "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave & Fullerton Ave" "Paulina St & Montrose Ave" ...  
## $ end\_station\_id : chr [1:5667717] "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...  
## $ start\_lat : num [1:5667717] 42 42 41.9 42 41.9 ...  
## $ start\_lng : num [1:5667717] -87.7 -87.7 -87.7 -87.7 -87.6 ...  
## $ end\_lat : num [1:5667717] 42 42 41.9 42 41.9 ...  
## $ end\_lng : num [1:5667717] -87.7 -87.7 -87.7 -87.7 -87.6 ...  
## $ member\_casual : chr [1:5667717] "casual" "casual" "member" "casual" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. ride\_id = col\_character(),  
## .. rideable\_type = col\_character(),  
## .. started\_at = col\_datetime(format = ""),  
## .. ended\_at = col\_datetime(format = ""),  
## .. start\_station\_name = col\_character(),  
## .. start\_station\_id = col\_character(),  
## .. end\_station\_name = col\_character(),  
## .. end\_station\_id = col\_character(),  
## .. start\_lat = col\_double(),  
## .. start\_lng = col\_double(),  
## .. end\_lat = col\_double(),  
## .. end\_lng = col\_double(),  
## .. member\_casual = col\_character()  
## .. )  
## - attr(\*, "problems")=<externalptr>

This provides the formats for the data read. As we can see, some columns are better off as factors

Columns we should have as factors are: rideable\_type and member\_casual

tripdata$rideable\_type <- as.factor(tripdata$rideable\_type)  
table(tripdata$rideable\_type)

##   
## classic\_bike docked\_bike electric\_bike   
## 2601214 177474 2889029

tripdata$member\_casual <- as.factor(tripdata$member\_casual)  
table(tripdata$member\_casual)

##   
## casual member   
## 2322032 3345685

OK, there are no spurious entries for these factors. The dates are ok. Are there NA values?

sum(is.na(tripdata$started\_at))

## [1] 0

sum(is.na(tripdata$ended\_at))

## [1] 0

sum(is.na(tripdata$ride\_id))

## [1] 0

sum(is.na(tripdata$rideable\_type))

## [1] 0

sum(is.na(tripdata$start\_lat))

## [1] 0

sum(is.na(tripdata$end\_lat))

## [1] 5858

sum(is.na(tripdata$start\_lng))

## [1] 0

sum(is.na(tripdata$end\_lng))

## [1] 5858

Only the end\_lat and end\_lng have an identical number of NAs. Why?

table(tripdata[is.na(tripdata$end\_lng),]$rideable\_type)

##   
## classic\_bike docked\_bike electric\_bike   
## 3242 2616 0

This really doesn’t say much.

## Creating new variables

We need to derive some additional data from the columns already present

1. Duration of the ride - **duration**
2. Date, extract from timestamp ‘started\_at’ We will ignore corner cases where rides were taken around midnight, and use only started\_at for the same - **date**
3. Weekday - Sunday, Monday etc. that the ride was taken - **day\_of\_the\_week** - a factor
4. Whether the ride was a weekday ride(Mon..Fri) or a weekend ride(Sat,Sun) - **weekday\_or\_end** - a factor
5. A combination of start\_station\_id and end\_station\_id - **commute**

For duration, it is simple. Luckily, the started\_at and ended\_at are both in POSIX\_ct format as timestamps.

tripdata$duration <- tripdata$ended\_at - tripdata$started\_at

sum(tripdata$duration<0)

## [1] 100

sum(tripdata$duration==0)

## [1] 431

Remove spurious data. Only duration>0 should be included. The 100 negative entries are likely errors in entry.

tripdata[tripdata$duration==0,]

## # A tibble: 431 × 14  
## ride\_id ridea…¹ started\_at ended\_at start…² start…³  
## <chr> <fct> <dttm> <dttm> <chr> <chr>   
## 1 C2E047DDF019… electr… 2022-01-18 19:25:42 2022-01-18 19:25:42 Green … TA1307…  
## 2 8D3E8E511FEB… electr… 2022-01-21 01:05:35 2022-01-21 01:05:35 Wester… 13068   
## 3 A753A729011B… electr… 2022-01-09 10:39:48 2022-01-09 10:39:48 Clark … KA1504…  
## 4 0C63D14D2612… electr… 2022-01-28 15:28:11 2022-01-28 15:28:11 Wells … TA1307…  
## 5 4B0FC5ACEE52… electr… 2022-01-18 19:38:26 2022-01-18 19:38:26 Sheffi… TA1309…  
## 6 6174209419E9… classi… 2022-02-17 14:20:58 2022-02-17 14:20:58 Univer… KA1503…  
## 7 FA080339DCF1… classi… 2022-02-12 15:07:50 2022-02-12 15:07:50 Michig… TA1305…  
## 8 2CF1A9AEEB7A… electr… 2022-02-17 09:39:01 2022-02-17 09:39:01 Despla… 15535   
## 9 21582FCE30A6… classi… 2022-02-16 14:32:32 2022-02-16 14:32:32 Canal … 13011   
## 10 CF76E9E68F30… classi… 2022-02-24 07:41:45 2022-02-24 07:41:45 Ashlan… 13224   
## # … with 421 more rows, 8 more variables: end\_station\_name <chr>,  
## # end\_station\_id <chr>, start\_lat <dbl>, start\_lng <dbl>, end\_lat <dbl>,  
## # end\_lng <dbl>, member\_casual <fct>, duration <drtn>, and abbreviated  
## # variable names ¹​rideable\_type, ²​start\_station\_name, ³​start\_station\_id

Its not clear why there are entries for 0 sec durations as they are either maintenance calls or errors. So we will remove all these:

tripdata <- tripdata[tripdata$duration>0,]

Date can easily be extracted, and the

tripdata$date <- lubridate::date(tripdata$started\_at)

From this, we get the day\_of\_the\_week

tripdata$day\_of\_the\_week <- as.factor(wday(tripdata$started\_at, label=TRUE))

We now need to decide whether the day is a weekday or weekend. This is simple:

weekday\_or\_end <- function(day) ifelse(day=="Sat"|day=="Sun","Weekend","Weekday")

We define the simple one-line function. Then we plug it into the assignment below. Note that these values should be factors

tripdata$weekday\_or\_end <- as.factor(weekday\_or\_end(as.character(tripdata$day\_of\_the\_week)))

One last check

str(tripdata)

## tibble [5,667,186 × 17] (S3: tbl\_df/tbl/data.frame)  
## $ ride\_id : chr [1:5667186] "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED419105406" ...  
## $ rideable\_type : Factor w/ 3 levels "classic\_bike",..: 3 3 1 1 1 1 1 1 3 1 ...  
## $ started\_at : POSIXct[1:5667186], format: "2022-01-13 11:59:47" "2022-01-10 08:41:56" ...  
## $ ended\_at : POSIXct[1:5667186], format: "2022-01-13 12:02:44" "2022-01-10 08:46:17" ...  
## $ start\_station\_name: chr [1:5667186] "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & Fullerton Ave" "Clark St & Bryn Mawr Ave" ...  
## $ start\_station\_id : chr [1:5667186] "525" "525" "TA1306000016" "KA1504000151" ...  
## $ end\_station\_name : chr [1:5667186] "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave & Fullerton Ave" "Paulina St & Montrose Ave" ...  
## $ end\_station\_id : chr [1:5667186] "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...  
## $ start\_lat : num [1:5667186] 42 42 41.9 42 41.9 ...  
## $ start\_lng : num [1:5667186] -87.7 -87.7 -87.7 -87.7 -87.6 ...  
## $ end\_lat : num [1:5667186] 42 42 41.9 42 41.9 ...  
## $ end\_lng : num [1:5667186] -87.7 -87.7 -87.7 -87.7 -87.6 ...  
## $ member\_casual : Factor w/ 2 levels "casual","member": 1 1 2 1 2 2 2 2 2 2 ...  
## $ duration : 'difftime' num [1:5667186] 177 261 261 896 ...  
## ..- attr(\*, "units")= chr "secs"  
## $ date : Date[1:5667186], format: "2022-01-13" "2022-01-10" ...  
## $ day\_of\_the\_week : Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<..: 5 2 3 3 5 3 1 7 2 6 ...  
## $ weekday\_or\_end : Factor w/ 2 levels "Weekday","Weekend": 1 1 1 1 1 1 2 2 1 1 ...

OK

# Analyse

## Extracting information

For analysis, we can generate a large number of graphs, only some of which may make sense, while others may give the same basic information. The various factor variables are member\_casual, weekday\_or\_end, rideable\_type and day\_of\_the\_week. All of these can be used for group\_by. The member\_casual factor is key, as the question to be answered revolves around it. So we will generate group-based analysis for each of the other factors. The value we need to check is duration, as it is the billable quantity here. We take sum, mean, mode, max, min and count.

### What day of the week

trips\_by\_weekday <- tripdata %>%  
 group\_by(day\_of\_the\_week, member\_casual,date) %>%  
 summarise(sum = sum(duration), mean = mean(duration), mode = mode(duration), max = max(duration), min = min(duration), count = sum(duration>10,na.rm=TRUE))

## `summarise()` has grouped output by 'day\_of\_the\_week', 'member\_casual'. You can  
## override using the `.groups` argument.

### Weekly trips on working day or weekend?

weekly\_trips <- tripdata %>%  
 group\_by(weekday\_or\_end, member\_casual,date) %>%  
 summarise(sum = sum(duration), mean = mean(duration), mode = mode(duration), max = max(duration), min = min(duration), count = sum(duration>10,na.rm=TRUE))

## `summarise()` has grouped output by 'weekday\_or\_end', 'member\_casual'. You can  
## override using the `.groups` argument.

### What type of bike is preferred?

bike\_prefs <- tripdata %>%  
 group\_by(rideable\_type, member\_casual,date) %>%  
 summarise(sum = sum(duration), mean = mean(duration), mode = mode(duration), max = max(duration), min = min(duration),count = sum(duration>10,na.rm=TRUE))

## `summarise()` has grouped output by 'rideable\_type', 'member\_casual'. You can  
## override using the `.groups` argument.

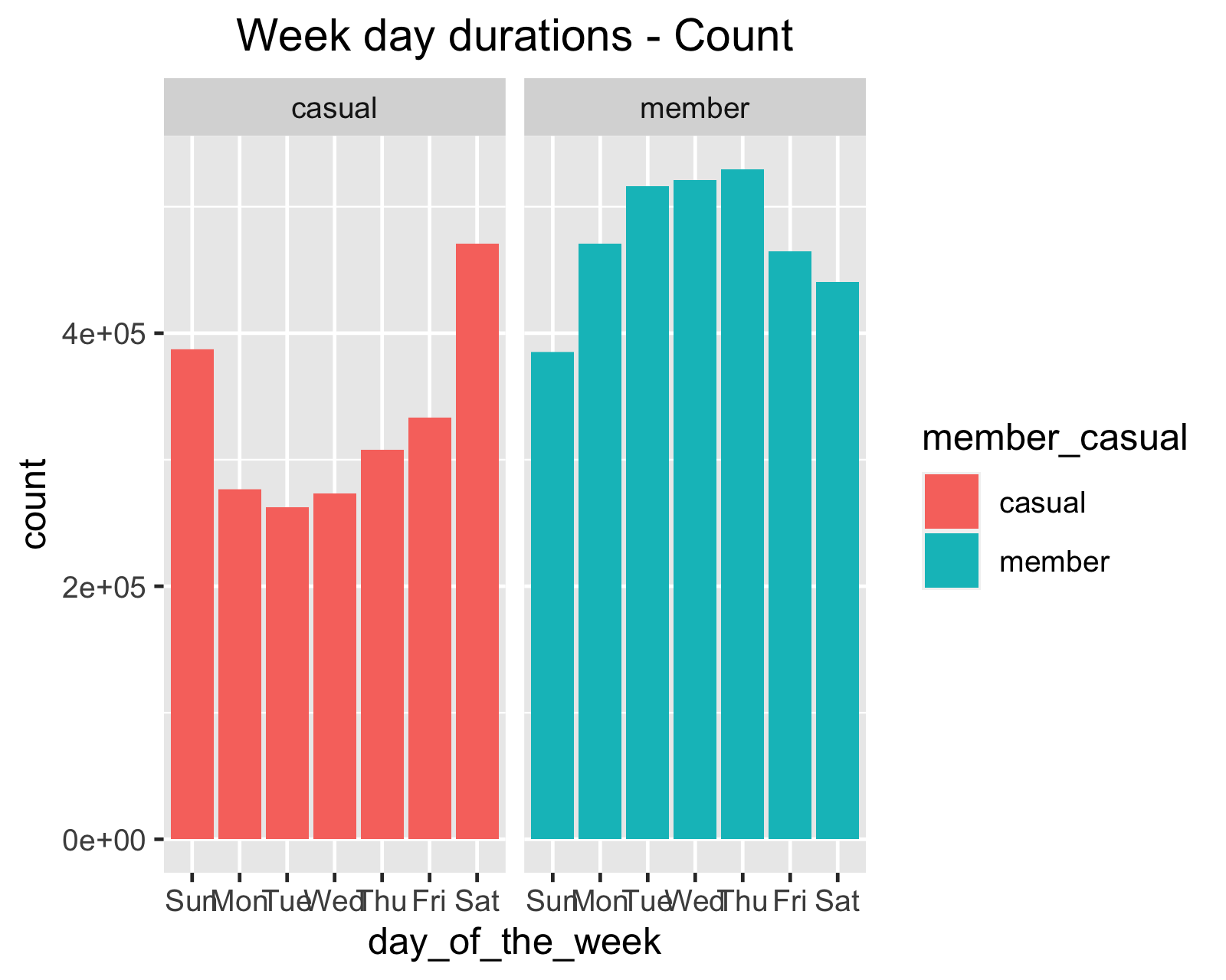
Many plots can be generated. But which of them make sense, which of them give us the information we need?

### Data for weekdays

#### Count

gg\_weekday\_bar <- ggplot(data = trips\_by\_weekday, aes(x=day\_of\_the\_week, y=count, fill=member\_casual)) +  
 facet\_grid(.~member\_casual) +  
 geom\_col()+  
 ggtitle("Week day durations - Count")+  
 theme(plot.title = element\_text(hjust = 0.5))  
ggsave(gg\_weekday\_bar,filename="gg\_weekday\_bar\_count.png",dpi=320)

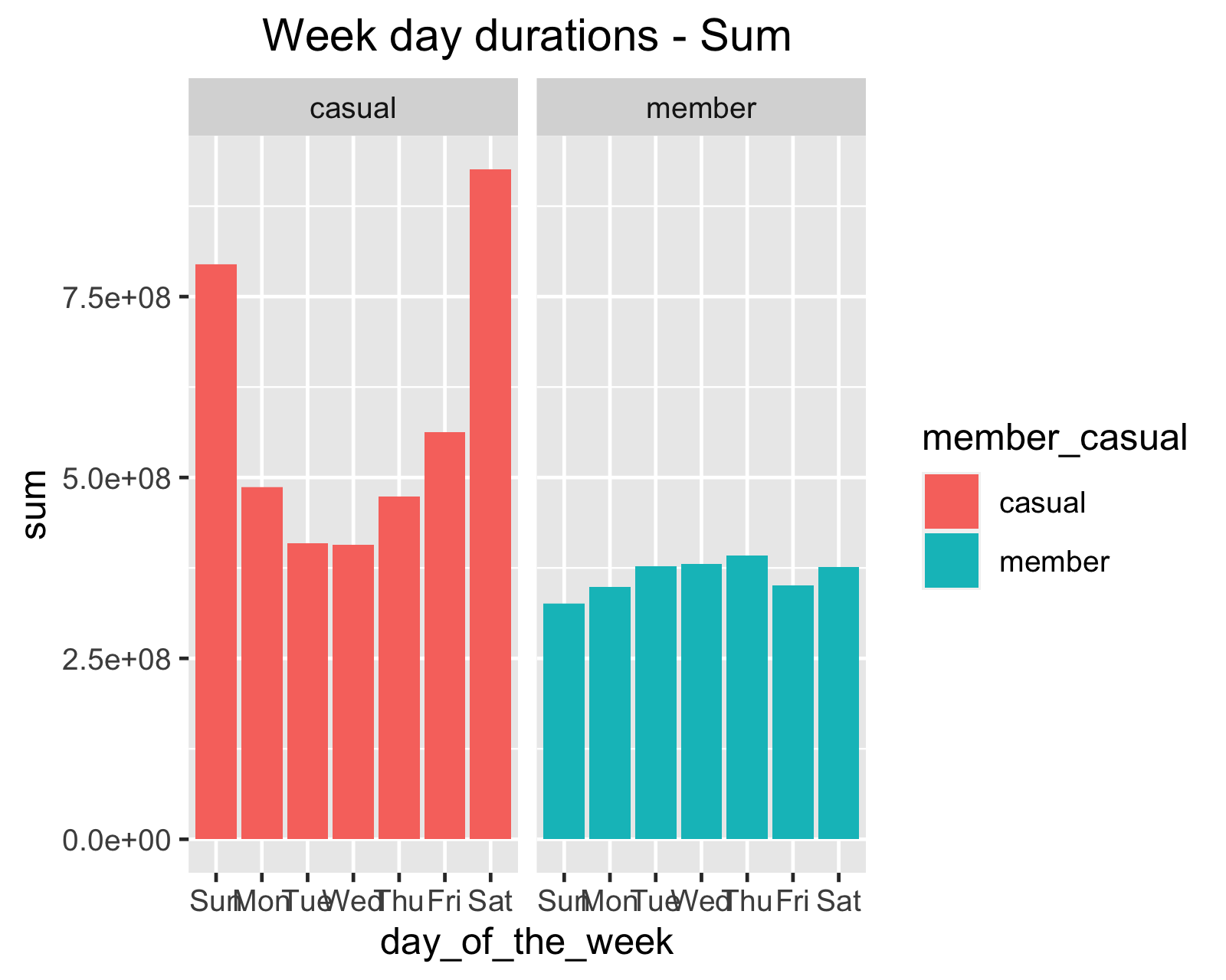
## Saving 5 x 4 in image



We can see that there are more member rides than casual rides. The casual riders are more in the weekends. We will see more weekday vs weekend information later. #### Sum

gg\_weekday\_bar <- ggplot(data = trips\_by\_weekday, aes(x=day\_of\_the\_week, y=sum, fill=member\_casual)) +  
 facet\_grid(.~member\_casual) +  
 geom\_col()+  
 ggtitle("Week day durations - Sum")+  
 theme(plot.title = element\_text(hjust = 0.5))  
ggsave(gg\_weekday\_bar,filename="gg\_weekday\_bar\_sum.png",dpi=320)

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.

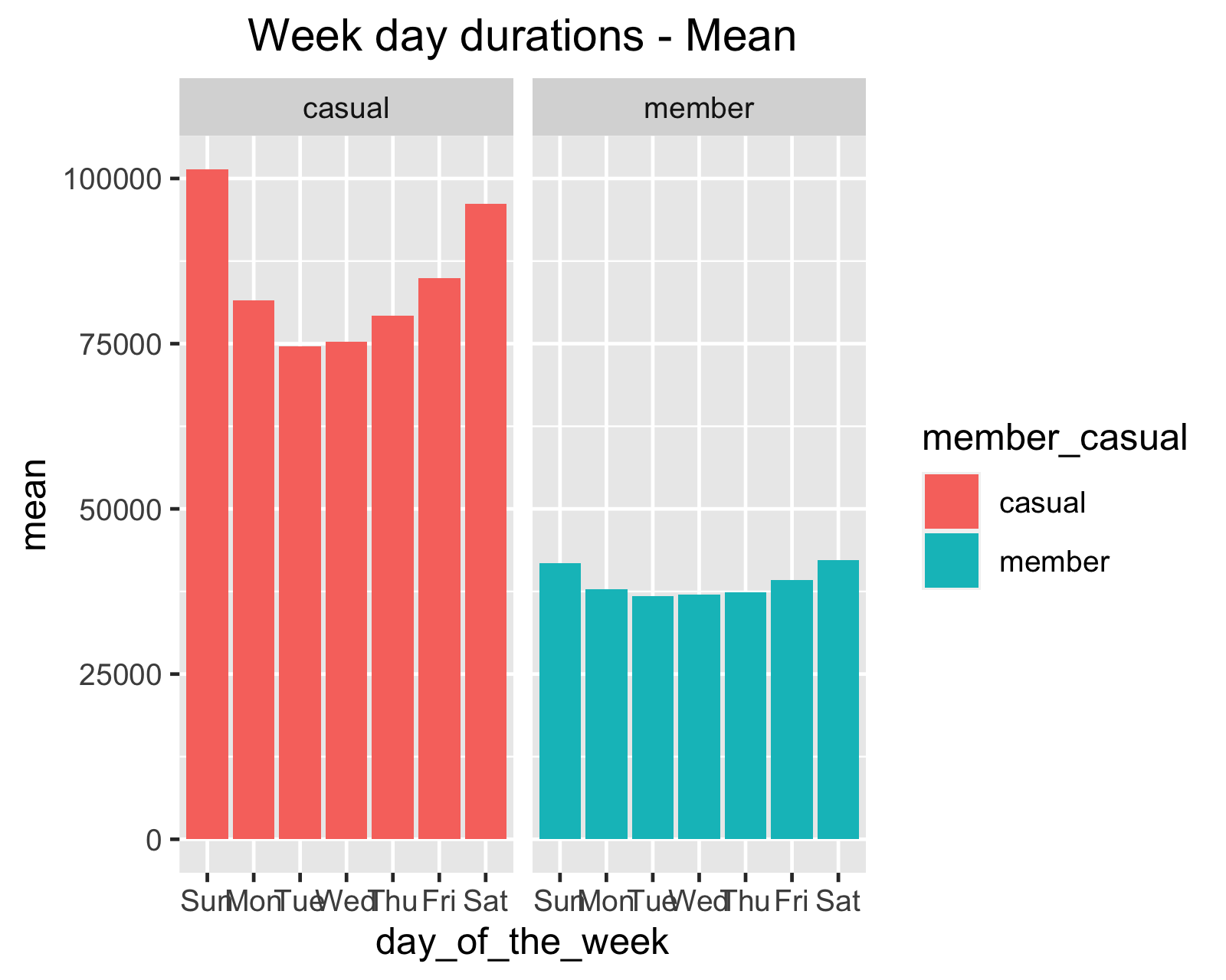


The pattern is even stronger here: in weekends, casual riders go out for long durations or many trips

#### Mean

gg\_weekday\_bar <- ggplot(data = trips\_by\_weekday, aes(x=day\_of\_the\_week, y=mean, fill=member\_casual)) +  
 facet\_grid(.~member\_casual) +  
 geom\_col()+  
 ggtitle("Week day durations - Mean")+  
 theme(plot.title = element\_text(hjust = 0.5))  
ggsave(gg\_weekday\_bar,filename="gg\_weekday\_bar\_mean.png",dpi=320)

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.

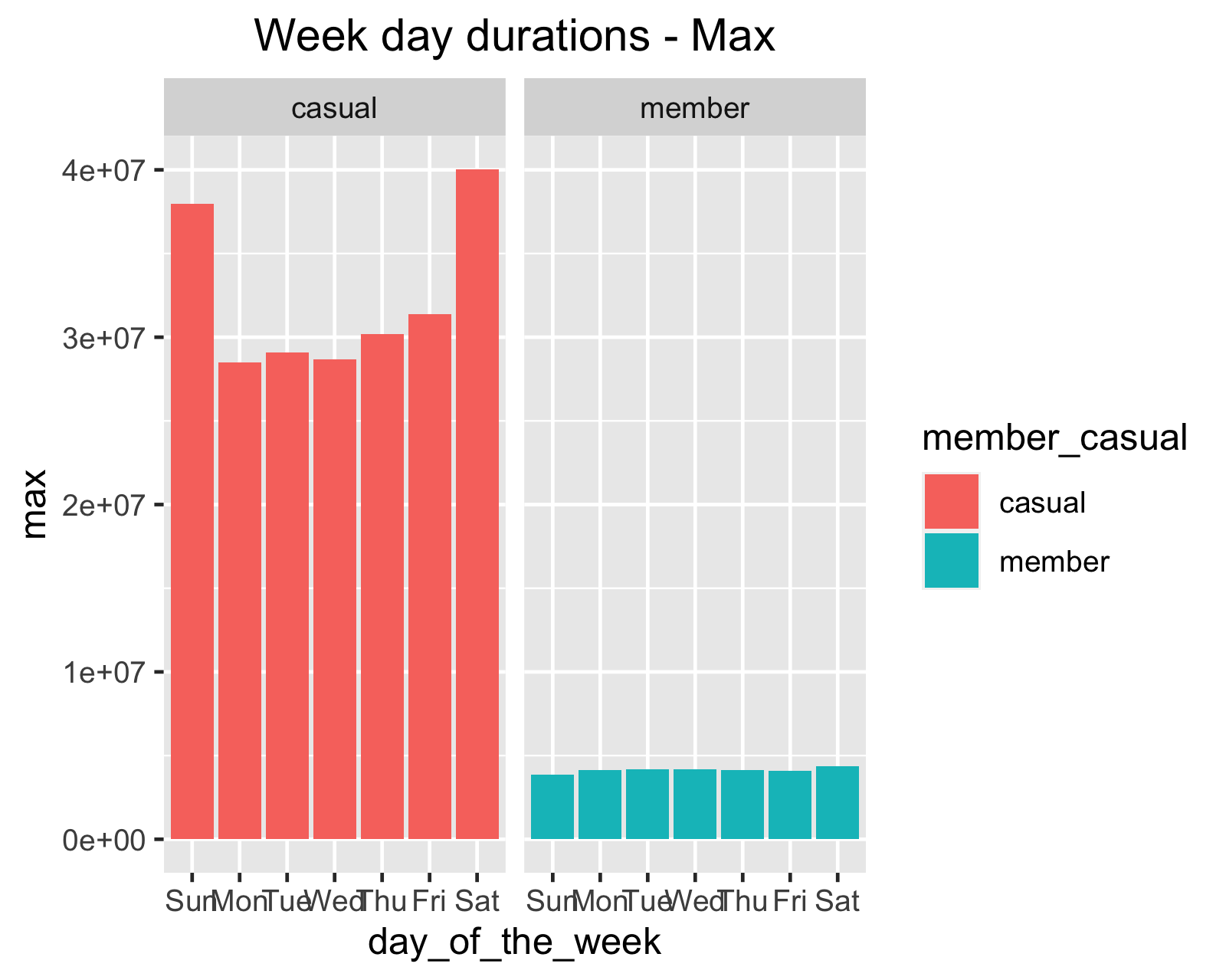


Looks like casual riders ride for longer durations. Lets confirm this:

#### Max

gg\_weekday\_bar <- ggplot(data = trips\_by\_weekday, aes(x=day\_of\_the\_week, y=max, fill=member\_casual)) +  
 facet\_grid(.~member\_casual) +  
 geom\_col()+  
 ggtitle("Week day durations - Max")+  
 theme(plot.title = element\_text(hjust = 0.5))  
ggsave(gg\_weekday\_bar,filename="gg\_weekday\_bar\_max.png",dpi=320)

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.



Looks like casual riders love to go on long rides. Not necessarily in the weekend, but certainly more. Member riders however seem to use these rides for fixed durations. How consistent in this are they?

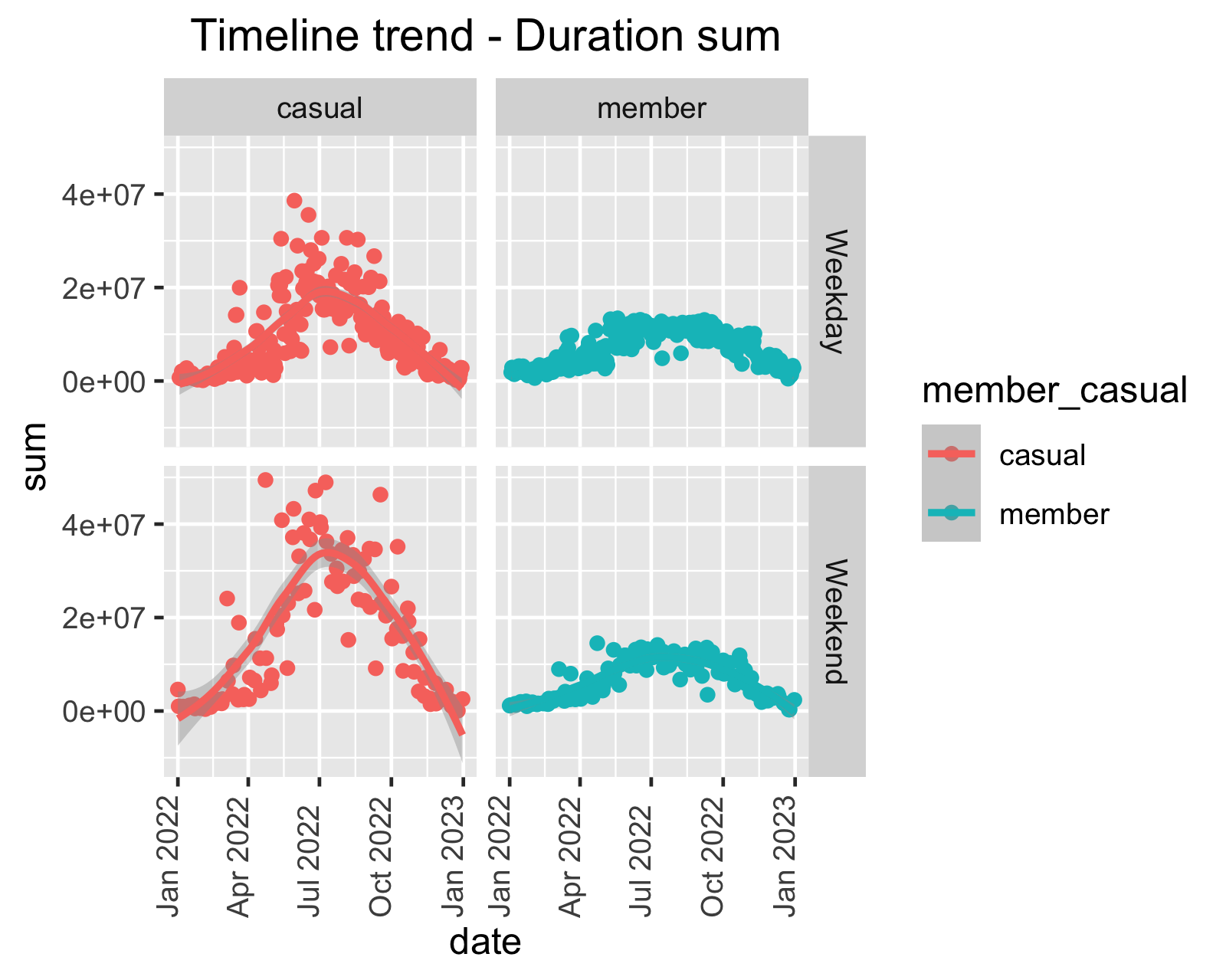
### Weekdays vs Weekends

We will now plot timelines for all the data through the year and get trends.

#### Sum

gg\_weekly\_sums <- ggplot(data = weekly\_trips, aes(x=date, y=sum, col=member\_casual)) +  
 facet\_grid(weekday\_or\_end~member\_casual) +  
 geom\_point() +  
 geom\_smooth()+  
 ggtitle("Timeline trend - Duration sum")+  
 theme(plot.title = element\_text(hjust = 0.5),axis.text.x = element\_text(angle = 90, vjust = 0, hjust=0))  
ggsave(gg\_weekly\_sums,filename="gg\_weekly\_sum.png",dpi=320)

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

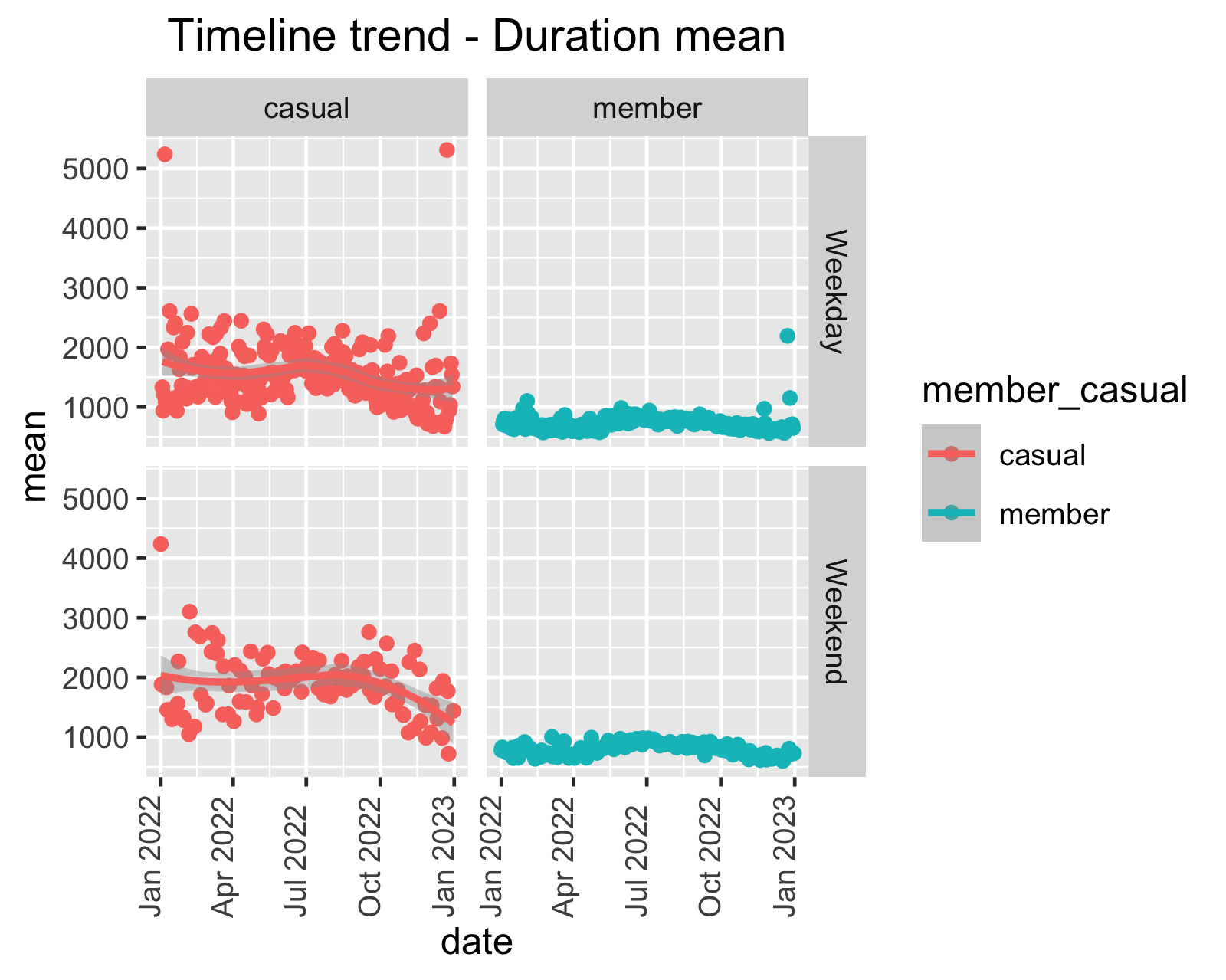


The members are much more consistent. Obviously the usage is more in the summer months.

#### Mean

gg\_weekly\_means <- ggplot(data = weekly\_trips, aes(x=date, y=mean, col=member\_casual)) +  
 facet\_grid(weekday\_or\_end~member\_casual) +  
 geom\_point() +  
 geom\_smooth()+  
ggtitle("Timeline trend - Duration mean")+  
 theme(plot.title = element\_text(hjust = 0.5),axis.text.x = element\_text(angle = 90, vjust = 0, hjust=0))  
ggsave(gg\_weekly\_means,filename="gg\_weekly\_means.png",dpi=320)

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

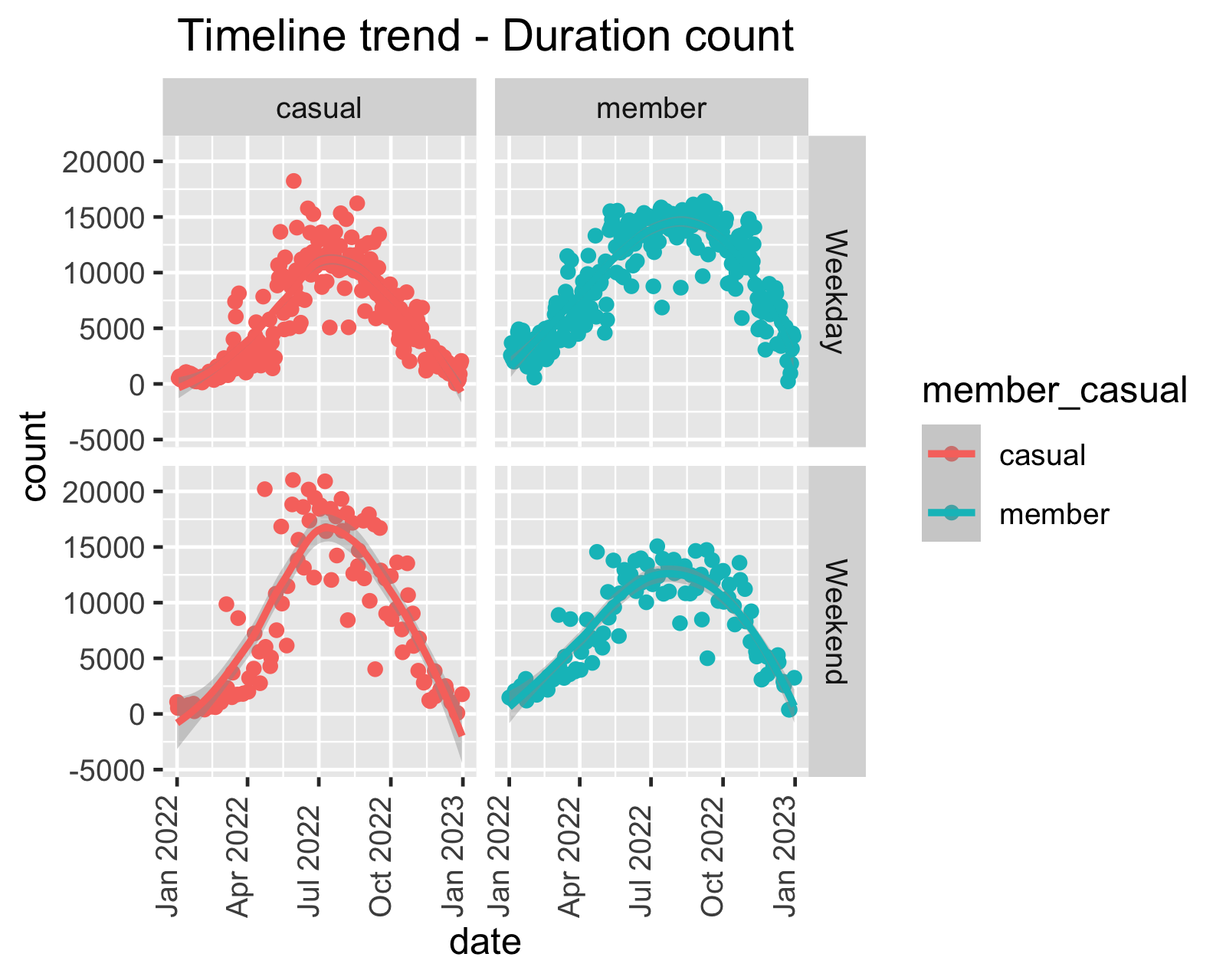


This strongly reinforces the consistency of the members

#### Count

gg\_weekly\_count <- ggplot(data = weekly\_trips, aes(x=date, y=count, col=member\_casual)) +  
 facet\_grid(weekday\_or\_end~member\_casual) +  
 geom\_point() +  
 geom\_smooth()+  
ggtitle("Timeline trend - Duration count")+  
 theme(plot.title = element\_text(hjust = 0.5),axis.text.x = element\_text(angle = 90, vjust = 0, hjust=0))  
ggsave(gg\_weekly\_count,filename="gg\_weekly\_count.png",dpi=320)

## Saving 5 x 4 in image  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

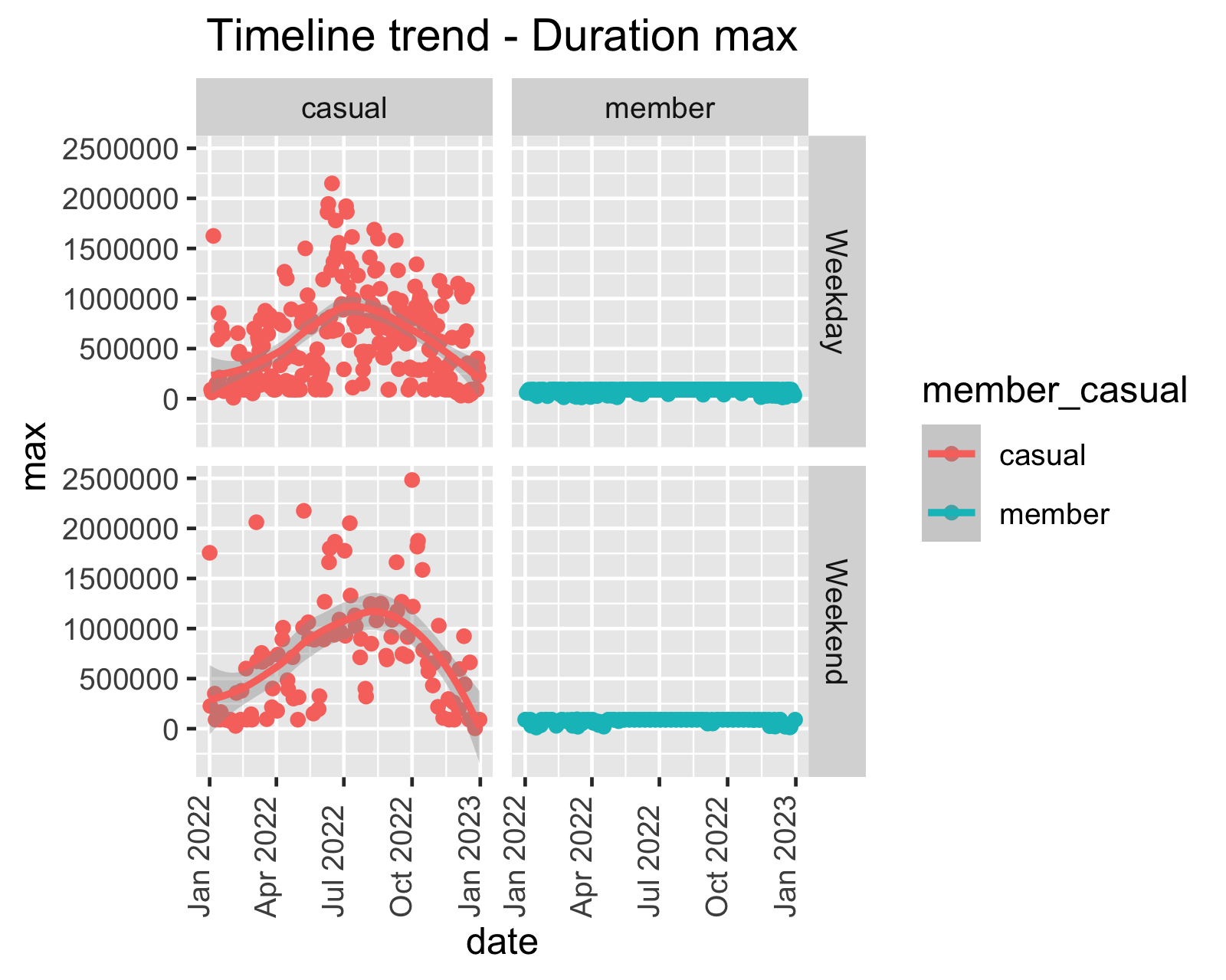


There is not much appreciable difference in counts. It seems members and casual riders take similar number of rides in the weekdays and weekends.

#### Max

gg\_week\_max <- ggplot(data = weekly\_trips, aes(x=date, y=max, col=member\_casual)) +  
 facet\_grid(weekday\_or\_end~member\_casual) +  
 geom\_point() +  
 geom\_smooth()+  
ggtitle("Timeline trend - Duration max")+  
 theme(plot.title = element\_text(hjust = 0.5),axis.text.x = element\_text(angle = 90, vjust = 0, hjust=0))  
ggsave(gg\_week\_max,filename="gg\_week\_max.png",dpi=320)

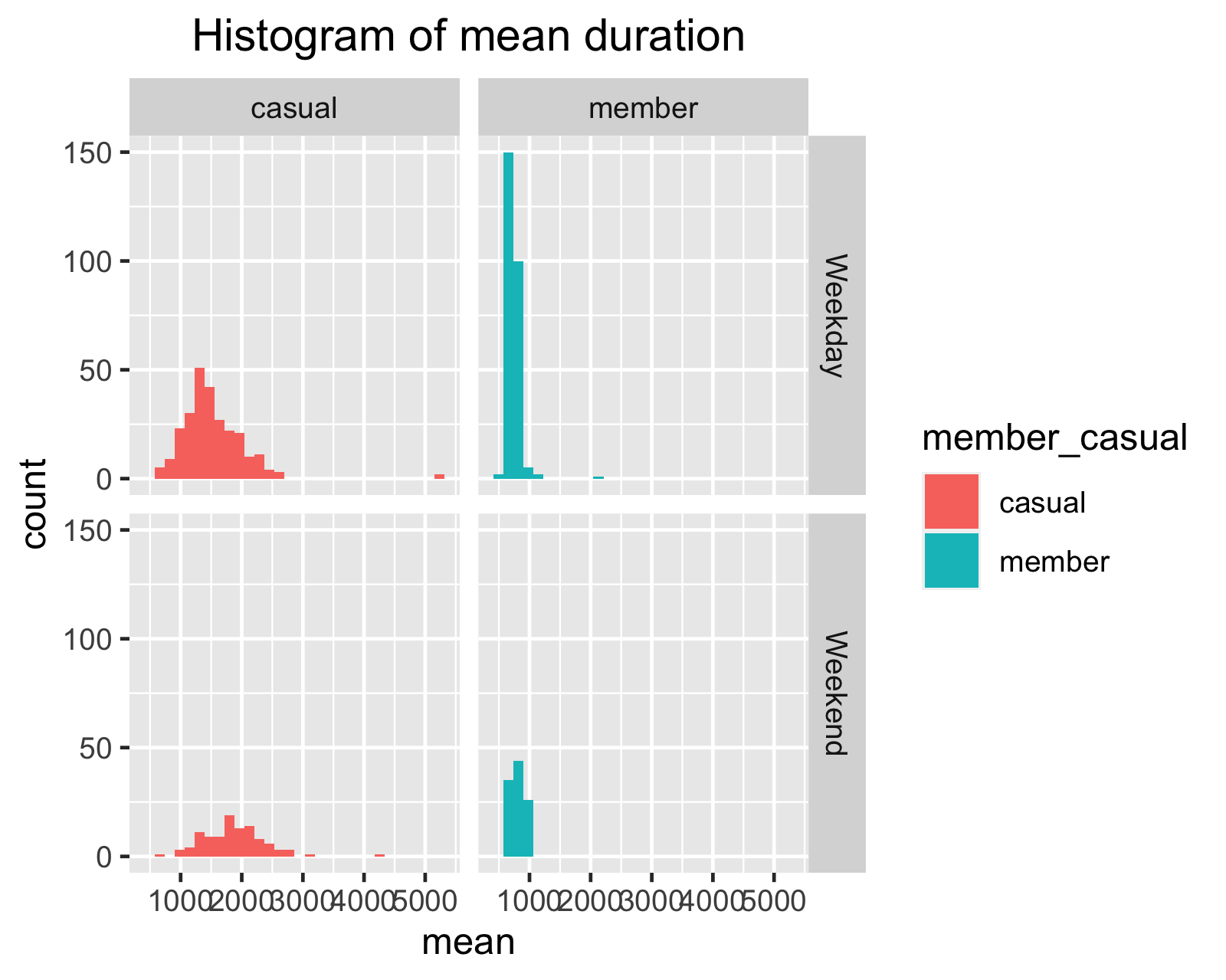
## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



This shows a drastic difference. The duration of member rides is highly restricted and those of casual riders are not. One obvious reason is that members use the rides for daily commute to work, while casual riders use them for adventure or recreation. To confirm this theory, lets take a histogram of means

gg\_weekday\_hist <- ggplot(data = weekly\_trips, aes(x=mean, fill=member\_casual)) +  
 facet\_grid(weekday\_or\_end~member\_casual) +  
 geom\_histogram()+  
 ggtitle("Histogram of mean duration")+  
 theme(plot.title = element\_text(hjust = 0.5))  
ggsave(gg\_weekday\_hist,filename="gg\_weekday\_hist.png",dpi=320)

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

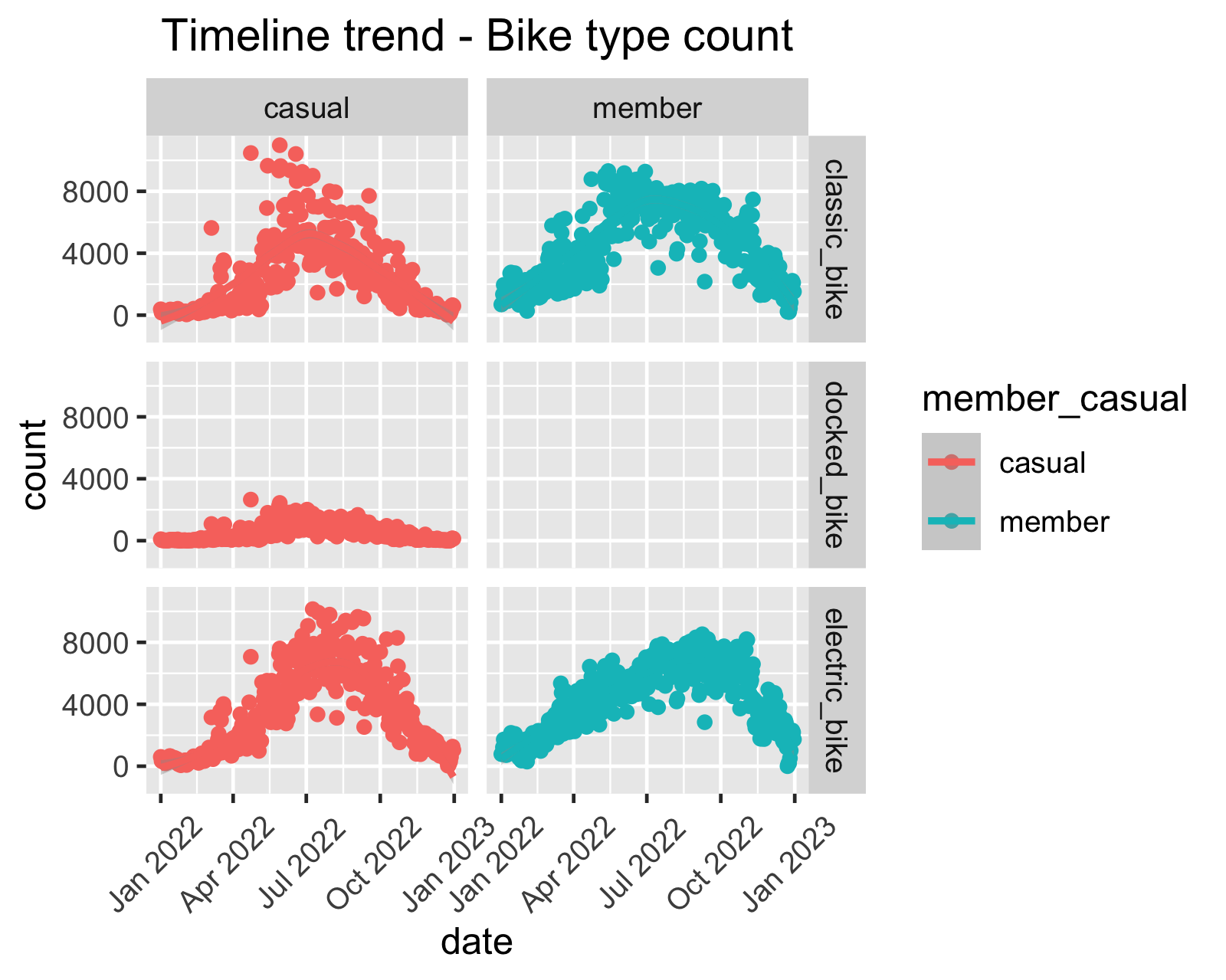


This tells us that short-distance commutes are what members prefer, while casual riders frequently go on long duration trips.

### Type of Bike

gg\_bike\_counts <- ggplot(data = bike\_prefs, aes(x=date, y=count, col=member\_casual)) +  
 facet\_grid(rideable\_type~member\_casual) +  
 geom\_point() +  
 geom\_smooth() +  
 ggtitle("Timeline trend - Bike type count")+  
 theme(plot.title = element\_text(hjust = 0.5),axis.text.x = element\_text(angle = 45, vjust = 0.6, hjust=0.6))  
ggsave(gg\_bike\_counts,filename="gg\_bike\_counts.png",dpi=320)

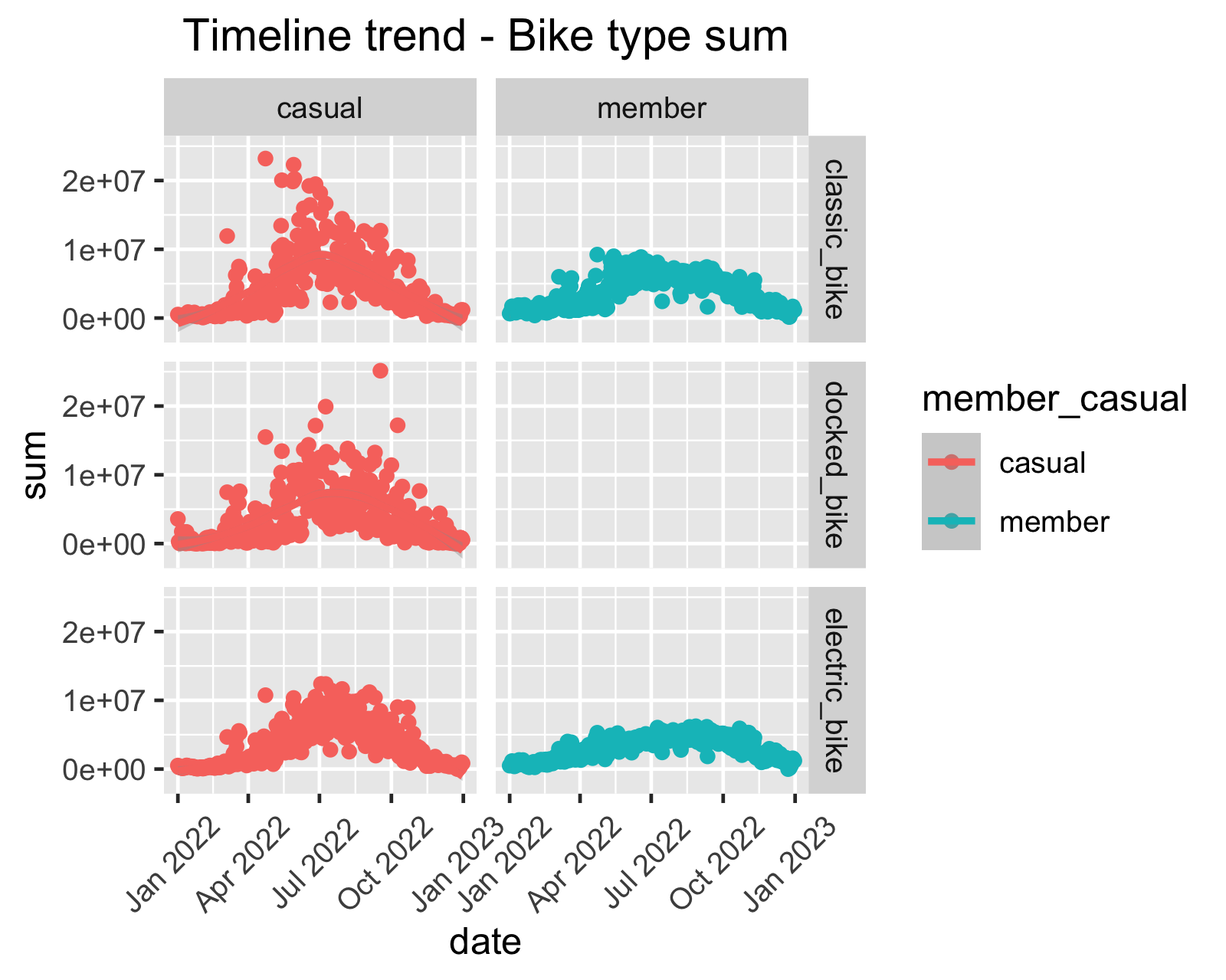
## Saving 5 x 4 in image  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



There is no obvious difference in usage of type of bike, though there is a preference by casual riders towards classic bike. There is no data for docked bike for members, docked bikes are used exclusively by casual users.

gg\_bike\_sum <- ggplot(data = bike\_prefs, aes(x=date, y=sum, col=member\_casual)) +  
 facet\_grid(rideable\_type~member\_casual) +  
 ggtitle("Timeline trend - Bike type sum")+  
 theme(plot.title = element\_text(hjust = 0.5),axis.text.x = element\_text(angle = 45, vjust = 0.6, hjust=0.6))+  
 geom\_point() +  
 geom\_smooth()  
  
ggsave(gg\_bike\_sum,filename="gg\_bike\_sum.png",dpi=320)

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



### Commute data

Lets check the type of rides that are preferred by casual vs members First lets remove NAs on start station and end station ids:

tripdata\_filtered <- tripdata[!is.na(tripdata$start\_station\_id) & !is.na(tripdata$end\_station\_id),]

Now, we make a combined variable of start station and end station. If start station and end station are identical, it means a round trip. We can have a boolean variable for that.

tripdata\_filtered$commute\_from\_to <- as.factor(paste(tripdata\_filtered$start\_station\_name,tripdata\_filtered$end\_station\_name,sep=".to."))  
tripdata\_filtered$commuteid\_from\_to <- as.factor(paste(tripdata\_filtered$start\_station\_id,tripdata\_filtered$end\_station\_id,sep="\_-\_"))  
tripdata\_filtered$roundtrip <- tripdata\_filtered$start\_station\_id == tripdata\_filtered$end\_station\_id

We can now get sum , mean, max and count elements for

tripdata\_filtered$commuteid\_from\_to <- as.factor(tripdata\_filtered$commuteid\_from\_to)  
  
commute\_prefs <- tripdata\_filtered %>%  
 group\_by(member\_casual,roundtrip) %>%  
 summarise(sum = sum(duration), mean=mean(duration),max=max(duration),count = sum(duration>10,na.rm=TRUE))

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

commute\_stats <- tripdata\_filtered %>%  
 group\_by(commuteid\_from\_to,member\_casual,roundtrip) %>%  
 summarise(count = sum(duration>10,na.rm=TRUE))

## `summarise()` has grouped output by 'commuteid\_from\_to', 'member\_casual'. You  
## can override using the `.groups` argument.

## since there are a large number of commuteid\_from\_to groups, don't use sum in summarise, or it takes forever

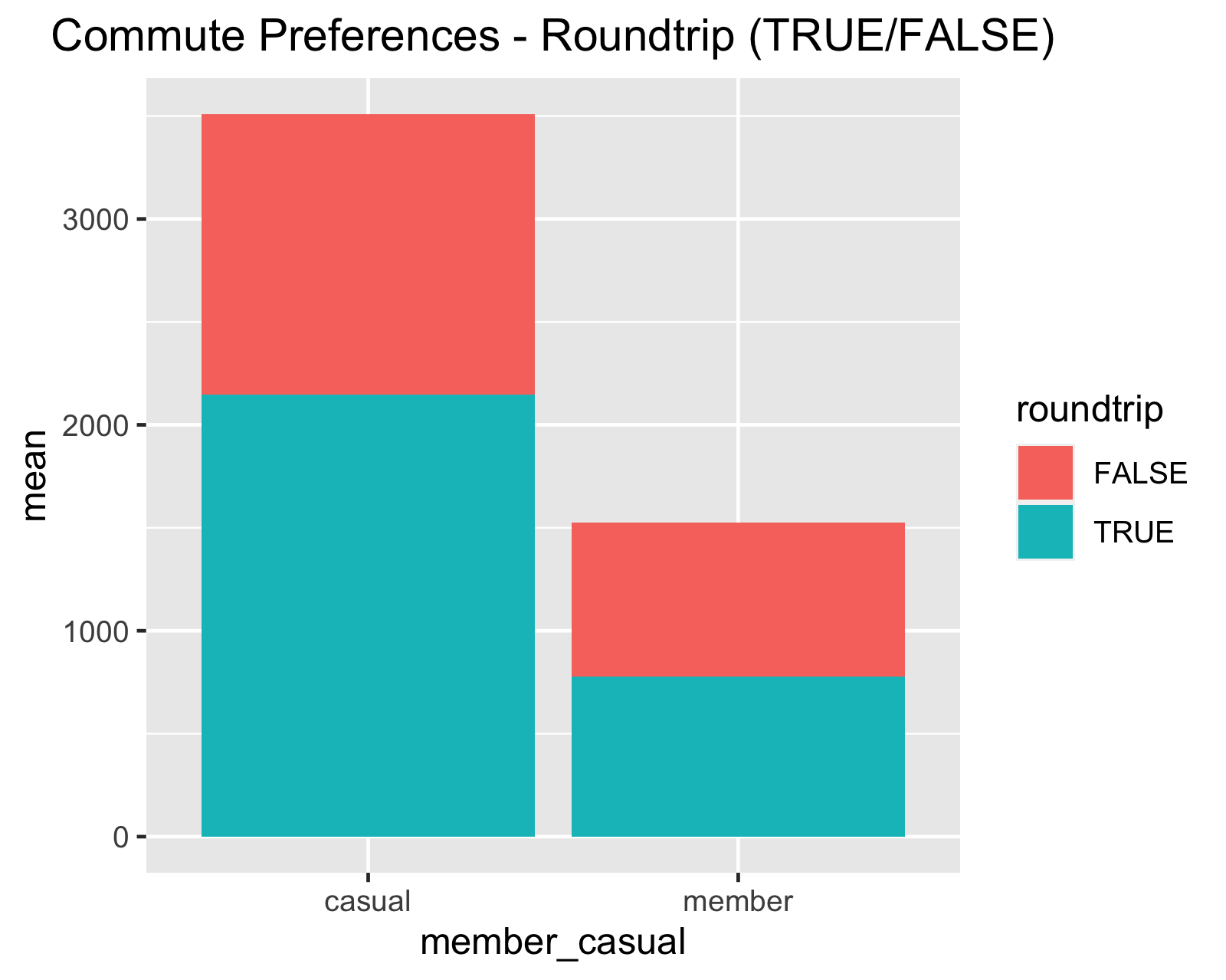
commute\_stats\_top <- commute\_stats %>%  
 arrange(desc(count)) %>%  
 group\_by(member\_casual) %>%  
 slice(1:10)  
commute\_stats\_top

## # A tibble: 20 × 4  
## # Groups: member\_casual [2]  
## commuteid\_from\_to member\_casual roundtrip count  
## <fct> <fct> <lgl> <int>  
## 1 13022\_-\_13022 casual TRUE 10447  
## 2 13300\_-\_13300 casual TRUE 6527  
## 3 13300\_-\_13022 casual FALSE 5100  
## 4 13042\_-\_13042 casual TRUE 4523  
## 5 13008\_-\_13008 casual TRUE 3987  
## 6 TA1308000012\_-\_TA1308000012 casual TRUE 2904  
## 7 13022\_-\_13300 casual FALSE 2854  
## 8 13022\_-\_13008 casual FALSE 2735  
## 9 15544\_-\_15544 casual TRUE 2423  
## 10 15544\_-\_13022 casual FALSE 2397  
## 11 KA1503000014\_-\_KA1503000071 member FALSE 5848  
## 12 KA1503000071\_-\_KA1503000014 member FALSE 5544  
## 13 KA1503000014\_-\_KA1504000076 member FALSE 5278  
## 14 KA1504000076\_-\_KA1503000014 member FALSE 4745  
## 15 13216\_-\_13217 member FALSE 3268  
## 16 13217\_-\_13216 member FALSE 3215  
## 17 13332\_-\_TA1307000130 member FALSE 2891  
## 18 TA1307000130\_-\_13332 member FALSE 2876  
## 19 KA1503000071\_-\_TA1309000037 member FALSE 2267  
## 20 13332\_-\_TA1307000121 member FALSE 2091

For casual riders, most of the rides are round trips, where destination is same as starting point.

gg\_commute\_bar <- ggplot(data = commute\_prefs, aes( x=member\_casual, y=mean, fill=roundtrip)) +  
# facet\_grid(.~roundtrip) +  
 geom\_col()+  
 ggtitle("Commute Preferences - Roundtrip (TRUE/FALSE)")+  
 theme(plot.title = element\_text(hjust = 0.5))  
ggsave(gg\_commute\_bar,filename="gg\_commute\_bar.png",dpi=320)

## Saving 5 x 4 in image  
## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.



This shows casual users prefer to use round trips for fairly long commutes.

## Synopsis of study

Using the above charts we can come to the conclusions that:-

1. Casual users use bikes more frequently in the weekends than in the weekdays
2. Casual users prefer long duration trips which are highly variable in duration, while members have fixed duration trips.
3. Member trip durations are very consistent - they indicate possible commute times to and from work. Casual users use rides likely for recreational activities or visits.
4. Casual members use more round trips. However, members also utilize round trips, though lesser. This may indicate errand runs during work.

## Share

Make a presentation of the data above with the above conclusions

## Act

1. Frequently used starting points for Casual users like station\_id 13022 (Streeter Dr & Grand Ave) or 13300 (DuSable Lake Shore Dr & Monroe St ) should have campaigns targeted at regular users frequenting these stations
2. Casual users who may use rides for daily commute should be made aware of the benefits of membership, and easy way to obtain the same
3. Weekday casual users should be made aware of the benefits of membership