Cyclistic customer analysis case study

Jacob Freed

May 29, 2023

Case study: How do the different types of Cyclistic customers use the bikes differently?

Welcome to my analysis of Cyclistic's data in R Markdown. I will be analyzing the provided dataset by Motivate International Inc.

This is analysis is for the Google Data Analytics Professional Certificate capstone project, supposing a fictional bike-sharing company to explore the data.

In this exercise, leadership has the following questions about this data:

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?

And I am going to answer those to the best of my ability using R Studio and will be documented in this RMarkdown file. Relevant files will be included alongside this file in my github repository https://github.com/Miles-Radium/Data-Analytics-Portfolio/.

Setting up and gathering insights

First we load up our libraries. As we have already installed them, I won't include the install code in this chunk.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(janitor)
```

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

```
library(knitr)
library(tidyr)
library(dplyr)
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(readr)
library(lubridate)
library(skimr)
```

Next we collate our data, in this case, the last 12 months of data into one dataframe for us to work with. The data in the directory can be added or removed as necessary to redo or modify the analysis as new information is added over time.

```
tripdata_last_12_months <- list.files(path = "/home/miles/Documents/school stuff & papers/Google Data An
                        pattern = "*.csv", full.names = TRUE) %>%
  lapply(read_csv, show_col_types = FALSE) %>%      # Store all files in list
  bind_rows                              # Combine all .CSVs into one data set
```

In order to ensure the data meshed together as it should have, let's get a look at its structure and the column names as well as the dimensions we are working with.

```
str(tripdata_last_12_months)
```

```
## spc_tbl_ [5,859,061 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5859061] "EC2DE40644C6B0F4" "1C31AD03897EE385" "1542FBEC830415CF" "6FF5
##  $ rideable_type     : chr [1:5859061] "classic_bike" "classic_bike" "classic_bike" "classic_bike" .
##  $ started_at        : POSIXct[1:5859061], format: "2022-05-23 23:06:58" "2022-05-11 08:53:28" ...
##  $ ended_at          : POSIXct[1:5859061], format: "2022-05-23 23:40:19" "2022-05-11 09:31:22" ...
##  $ start_station_name: chr [1:5859061] "Wabash Ave & Grand Ave" "DuSable Lake Shore Dr & Monroe St"
```

```
##  $ start_station_id  : chr [1:5859061] "TA1307000117" "13300" "TA1305000032" "TA1305000032" ...
##  $ end_station_name  : chr [1:5859061] "Halsted St & Roscoe St" "Field Blvd & South Water St" "Wood
##  $ end_station_id    : chr [1:5859061] "TA1309000025" "15534" "13221" "TA1305000030" ...
##  $ start_lat         : num [1:5859061] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:5859061] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:5859061] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:5859061] -87.6 -87.6 -87.7 -87.6 -87.7 ...
##  $ member_casual     : chr [1:5859061] "member" "member" "member" "member" ...
##  - 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>
```

```
colnames(tripdata_last_12_months)
```

```
##  [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"
```

```
dim(tripdata_last_12_months)
```

```
## [1] 5859061      13
```

To better understand the information we are working with, I like to run a few commands to get an idea about it.

```
# Getting a glimpse of the information we are working with.
glimpse(tripdata_last_12_months)
```

```
## Rows: 5,859,061
## Columns: 13
## $ ride_id            <chr> "EC2DE40644C6B0F4", "1C31AD03897EE385", "1542FBEC83~
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl~
## $ started_at         <dttm> 2022-05-23 23:06:58, 2022-05-11 08:53:28, 2022-05-~
## $ ended_at           <dttm> 2022-05-23 23:40:19, 2022-05-11 09:31:22, 2022-05-~
## $ start_station_name <chr> "Wabash Ave & Grand Ave", "DuSable Lake Shore Dr & ~
## $ start_station_id   <chr> "TA1307000117", "13300", "TA1305000032", "TA1305000~
```

```
## $ end_station_name   <chr> "Halsted St & Roscoe St", "Field Blvd & South Water~
## $ end_station_id     <chr> "TA1309000025", "15534", "13221", "TA1305000030", "~
## $ start_lat          <dbl> 41.89147, 41.88096, 41.88224, 41.88224, 41.88224, 4~
## $ start_lng          <dbl> -87.62676, -87.61674, -87.64107, -87.64107, -87.641~
## $ end_lat            <dbl> 41.94367, 41.88635, 41.90765, 41.88458, 41.88578, 4~
## $ end_lng            <dbl> -87.64895, -87.61752, -87.67255, -87.63189, -87.651~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
# Scanning the top of the dataset
head(tripdata_last_12_months)
```

```
## # A tibble: 6 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 EC2DE40644C6B0F4 classic_bike  2022-05-23 23:06:58 2022-05-23 23:40:19
## 2 1C31AD03897EE385 classic_bike  2022-05-11 08:53:28 2022-05-11 09:31:22
## 3 1542FBEC830415CF classic_bike  2022-05-26 18:36:28 2022-05-26 18:58:18
## 4 6FF59852924528F8 classic_bike  2022-05-10 07:30:07 2022-05-10 07:38:49
## 5 483C52CAAE12E3AC classic_bike  2022-05-10 17:31:56 2022-05-10 17:36:57
## 6 C0A3AA5A614DCE01 classic_bike  2022-05-04 14:48:55 2022-05-04 14:56:04
## # i 9 more variables: 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>
```

```
# Skimming gets us another idea of the datasets
skim_without_charts(tripdata_last_12_months)
```

Table 1: Data summary

Name	tripdata_last_12_months
Number of rows	5859061
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	1.00	16	16	0	5859061	0
rideable_type	0	1.00	11	13	0	3	0
start_station_name	832009	0.86	3	64	0	1722	0
start_station_id	832141	0.86	3	36	0	1319	0
end_station_name	889661	0.85	3	64	0	1741	0
end_station_id	889802	0.85	3	36	0	1324	0
member_casual	0	1.00	6	6	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
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	5973	1	41.90	0.07	0.00	41.88	41.90	41.93	42.37
end_lng	5973	1	-87.65	0.11	-88.14	-87.66	-87.64	-87.63	0.00

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2022-05-01 00:00:06	2023-04-30 23:59:05	2022-08-28 12:44:57	4916326
ended_at	0	1	2022-05-01 00:05:17	2023-05-03 10:37:12	2022-08-28 13:07:09	4930169

```
summary(tripdata_last_12_months) # Summary of the information
```

```
##    ride_id          rideable_type        started_at                 
##  Length:5859061     Length:5859061     Min.   :2022-05-01 00:00:06  
##  Class :character   Class :character   1st Qu.:2022-07-03 11:12:30  
##  Mode  :character   Mode  :character   Median :2022-08-28 12:44:57  
##                                        Mean   :2022-09-19 13:39:54  
##                                        3rd Qu.:2022-11-08 06:30:21  
##                                        Max.   :2023-04-30 23:59:05  
##                                                                     
##     ended_at                      start_station_name start_station_id  
##  Min.   :2022-05-01 00:05:17   Length:5859061     Length:5859061    
##  1st Qu.:2022-07-03 11:38:52   Class :character   Class :character  
##  Median :2022-08-28 13:07:09   Mode  :character   Mode  :character  
##  Mean   :2022-09-19 13:58:50                                        
##  3rd Qu.:2022-11-08 06:43:39                                        
##  Max.   :2023-05-03 10:37:12                                        
##                                                                     
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:5859061     Length:5859061     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.   : 0.00   Min.   :-88.14   Length:5859061    
##  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.   :42.37   Max.   :  0.00                     
##  NA's   :5973    NA's   :5973
```

My analysis would probably be aided by taking the started and ended times and mutating a column of total elapsed time the bicycle was in use.

```
tripdata_last_12_months <-
  mutate(tripdata_last_12_months, total_ride_duration =
  difftime(tripdata_last_12_months$ended_at, tripdata_last_12_months$started_at))
```

Data cleaning and renaming

With a convenient column of data on hand to be compared against, we can clean the data of observations with impossible times, namely those with 0 or negative durations.

```
time_errors <- tripdata_last_12_months %>%
  filter(total_ride_duration <= 0) # Collected for future reference
print(paste("Erroneous observations: ", nrow(time_errors)))
```

```
## [1] "Erroneous observations:  544"
```

```
tripdata_last_12_months <- tripdata_last_12_months %>%
  filter(total_ride_duration > 0) # Removed from our working dataset
```

Renaming the variable details should clear up their usage for the analyses we'll perform.

```
tripdata_last_12_months <- tripdata_last_12_months %>%
  rename(bicycle_type = rideable_type, customers = member_casual,
         started_ride_at = started_at, ended_ride_at = ended_at)
```

```
tripdata_last_12_months$customers <-
  gsub("member", "annual member", tripdata_last_12_months$customers)
tripdata_last_12_months$customers <-
  gsub("casual", "casual rider", tripdata_last_12_months$customers)
```

Let's add a few more columns to get the full effect from the analysis. Seeing as we only have the times the ride started and concluded with, I will take those and turn them in a more robust set of time data for our analysis.

```
tripdata_last_12_months$date <-
  as.Date(tripdata_last_12_months$started_ride_at) # Human-readable date
tripdata_last_12_months$month <-
  month(as.Date(tripdata_last_12_months$date), label = TRUE)
  # The month in words rather than the number
tripdata_last_12_months <- tripdata_last_12_months %>%
  mutate(day = wday(date, label = TRUE)) # The day of the week in their names.
tripdata_last_12_months$hour <-
  (format(tripdata_last_12_months$started_ride_at,format="%H"))
  # Hour from 0 to 23
```

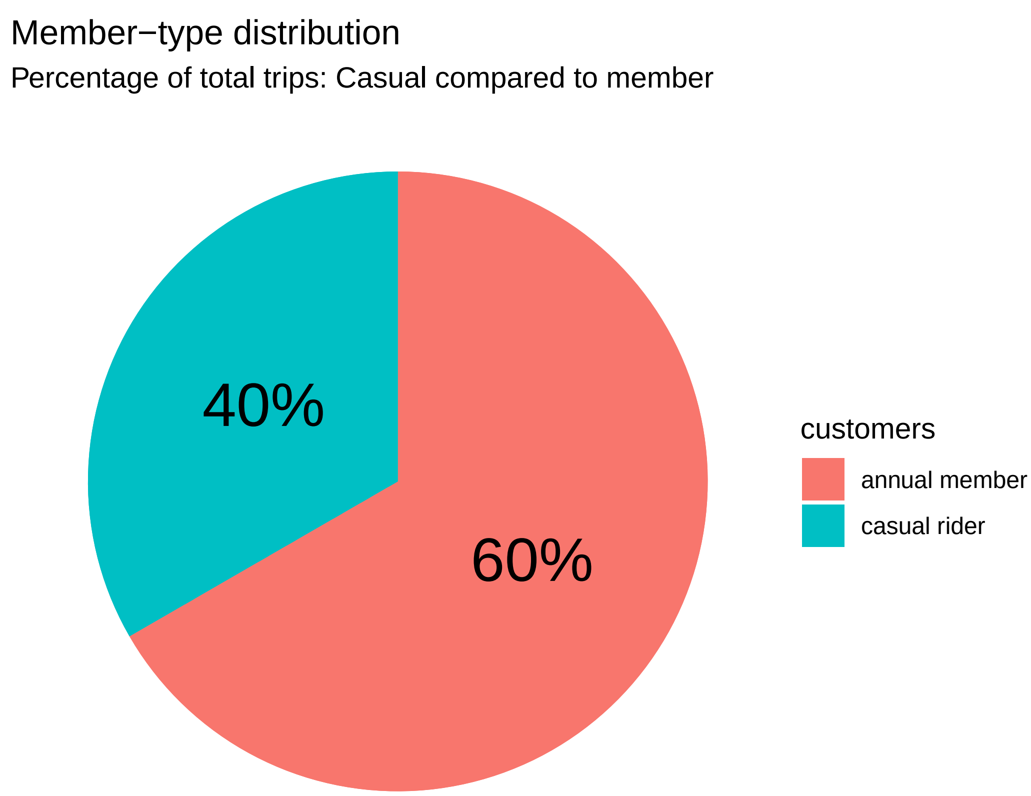
Analysis

Our understanding might be improved by understanding the proportion of member vs casual trips side by side.

```
tripdata_last_12_months %>%
  group_by(customers) %>%
  summarize("Number of trips" = n(),
            "Total trips" = nrow(tripdata_last_12_months),
            "% of total trips" = percent(n()/nrow(tripdata_last_12_months)))
```

```
## # A tibble: 2 x 4
##   customers     `Number of trips` `Total trips` `% of total trips`
##   <chr>                     <int>         <int> <chr>
## 1 annual member           3500469       5858517 60%
## 2 casual rider            2358048       5858517 40%
```

```
            # Getting numbers directly
tripdata_last_12_months %>% group_by(customers) %>%
  summarize("Number of trips" = n(),
            "Total trips" = nrow(tripdata_last_12_months),
            "Percentage" = percent(n()/nrow(tripdata_last_12_months))) %>%
  ggplot(aes(x="", y= Percentage, fill=customers)) + geom_col() +
  geom_bar(stat="identity") + coord_polar(theta = "y") + theme_void() +
  geom_text(aes(label = Percentage), color = "black", size=8,
            position = position_stack(vjust = 0.5)) +
  labs(title="Member-type distribution",
       subtitle = "Percentage of total trips: Casual compared to member")
```



```
# Visualizations really get the picture across
```

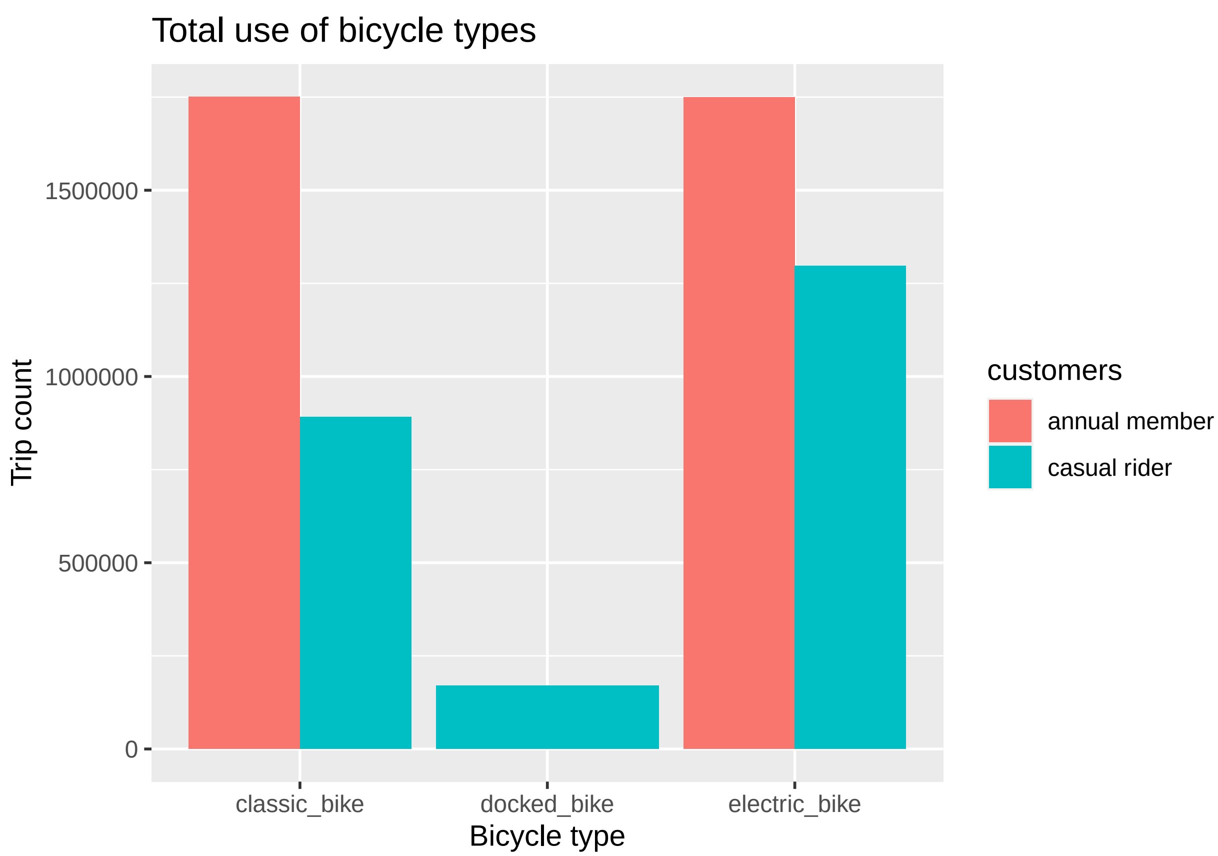
Above, we can see both visually and in the table, 60% of the riders are already members. Now let's see how bike use varies by customer type.

```
table(tripdata_last_12_months$bicycle_type)
```

```
##
##   classic_bike   docked_bike electric_bike
##        2642461        170513       3045543
```

```
tripdata_last_12_months %>% group_by(customers, bicycle_type) %>%
  summarize(number_of_trips=n()) %>%
  ggplot(aes(x=bicycle_type, y=number_of_trips, fill = customers)) +
  geom_col(position="dodge") +
  labs(title="Total use of bicycle types", x="Bicycle type", y="Trip count")
```

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



This gets us one step nearer to answering how annual members and casual riders use the bikes differently. You can see a total absence of docked bicycle use for the annual members as well as a preference for electric bicycles for casual riders.

Now, let's see if there is any patterns to their ride times.

```
tripdata_last_12_months %>%  group_by(customers) %>%
  summarise("Number of trips" = n(),
            "Average trip time" = mean(total_ride_duration),
            "Minimum trip time" = min(total_ride_duration),
            "Median trip time" = median(total_ride_duration),
            "Maximum trip time" = max(total_ride_duration))
```

```
## # A tibble: 2 x 6
##   customers     `Number of trips` `Average trip time` `Minimum trip time`
##   <chr>                     <int> <drtn>              <drtn>
## 1 annual member           3500469  750.0693 secs      1 secs
## 2 casual rider            2358048 1709.8282 secs      1 secs
## # i 2 more variables: `Median trip time` <drtn>, `Maximum trip time` <drtn>
```

Now we are getting somewhere. We can see the annual members spend much less time riding and that is reflected in their lower average, median, and even maximum trip times. 43.8% of the average trip time compared to casuals, 69.2% of the median trip time vs casuals, and a measly 3.7% compared against casuals for maximum. Good to know.

```
750.0693 / 1709.8282 * 100
```

```
## [1] 43.86811
```

```
520 / 751 * 100
```

```
## [1] 69.24101
```

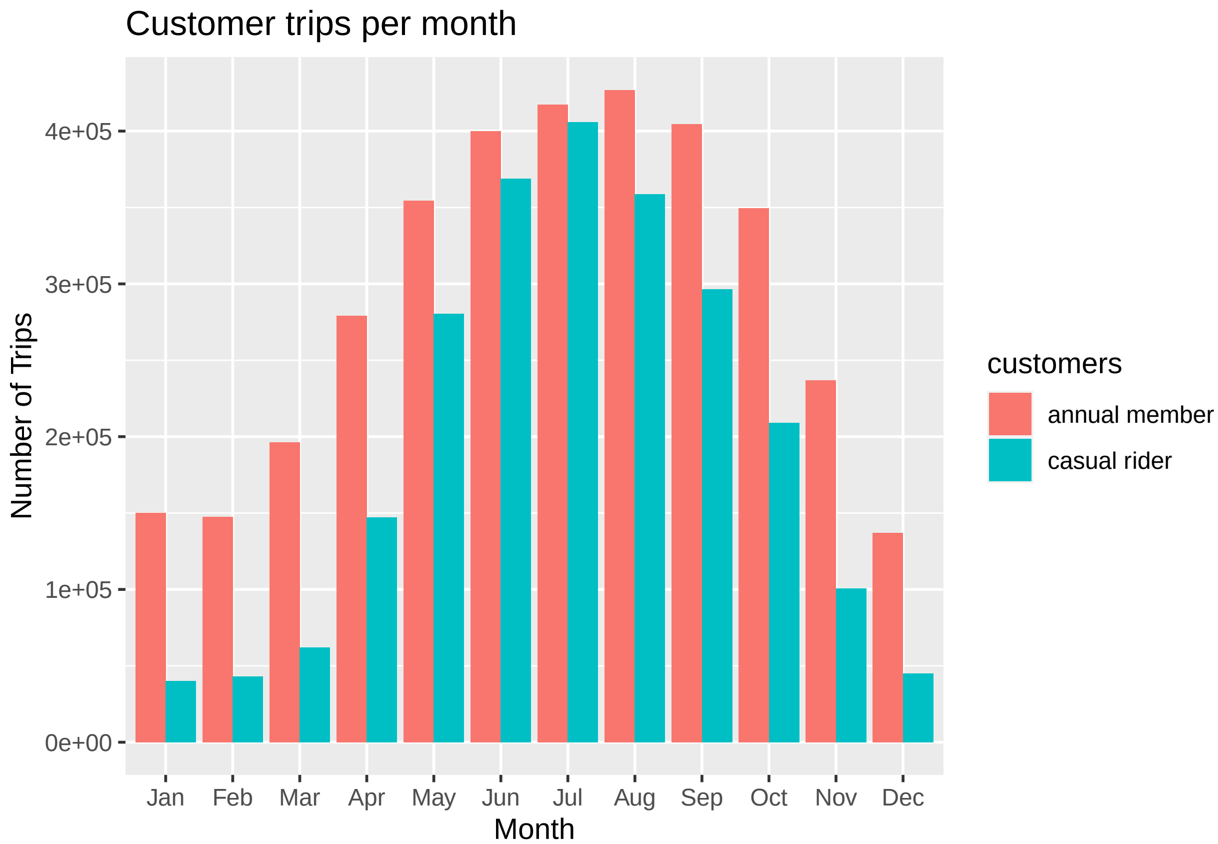
```
93580 / 2483235 * 100
```

```
## [1] 3.768471
```

As the data demonstrates, total bike sharing usage drops in the winter months (November to February) and correspondingly increases in the summer months (May to September). This effect is particularly pronounced for the casual riders. June and July almost sees parity between the customer types. Annual member use peaks in August, as compared to July for casuals.

```
ggplot(tripdata_last_12_months %>%
         group_by(customers, month) %>%
         summarise(number_of_rides = n())) +
  geom_col(position="dodge",
           mapping= aes(x = month, y = number_of_rides, fill = customers)) +
  labs(title="Customer trips per month",x = "Month", y = "Number of Trips")
```

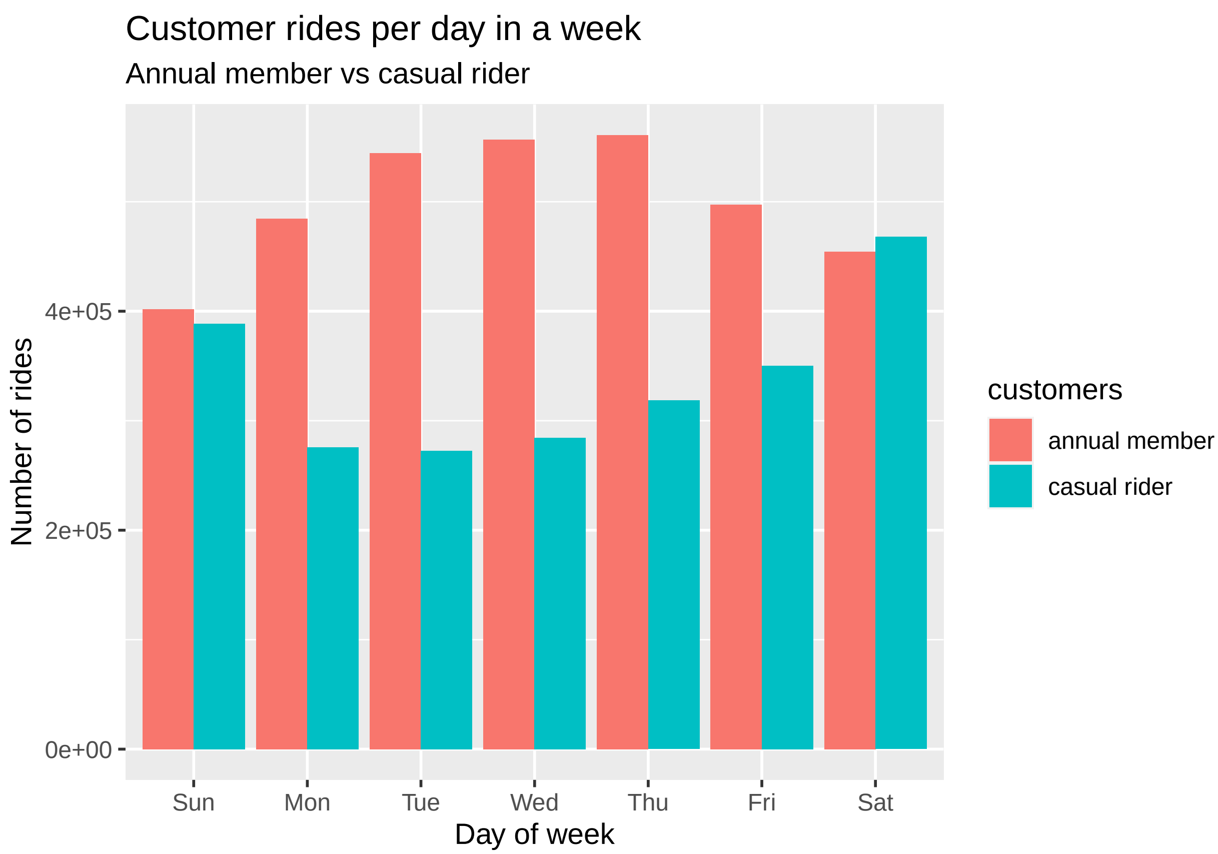
```
## `summarise()` has grouped output by 'customers'. You can override using the
## `.groups` argument.
```



Here's where the data gets interesting, member bike use spikes in the middle of the week and is at the lowest during the weekend while the opposite is true for the casual customers. This implies that the annual members use this service to get to and from work, while the casuals are more likely to be sightseers and others who bike on weekends.

```
ggplot(tripdata_last_12_months %>% group_by(customers, day) %>%
         summarise(number_of_rides = n())) +
  geom_col(position="dodge",
           mapping= aes(x = day, y = number_of_rides, fill = customers)) +
  labs(title="Customer rides per day in a week",
       subtitle = "Annual member vs casual rider",
       x = "Day of week", y = "Number of rides")
```

```
## ‘summarise()‘ has grouped output by ’customers’. You can override using the
## ‘.groups‘ argument.
```



Our understanding of the customers builds with this graph of the trip durations. The member durations stay close to each other the whole week, from ~12 minutes to approximately 14 minutes, while the casual riders reach a valley in the middle of the week and reach their apex on the weekend, ranging from about 24 minutes at the lowest to ~34.5 at the zenith. Casual trips still make up less of the rides, but they run for over double the trip time for their member counterparts, likely meaning they travel a considerably larger distance.

```
tripdata_last_12_months %>%  group_by(customers, day) %>%
  summarise(number_of_trips = n(), average_trip_time = mean(total_ride_duration))
```

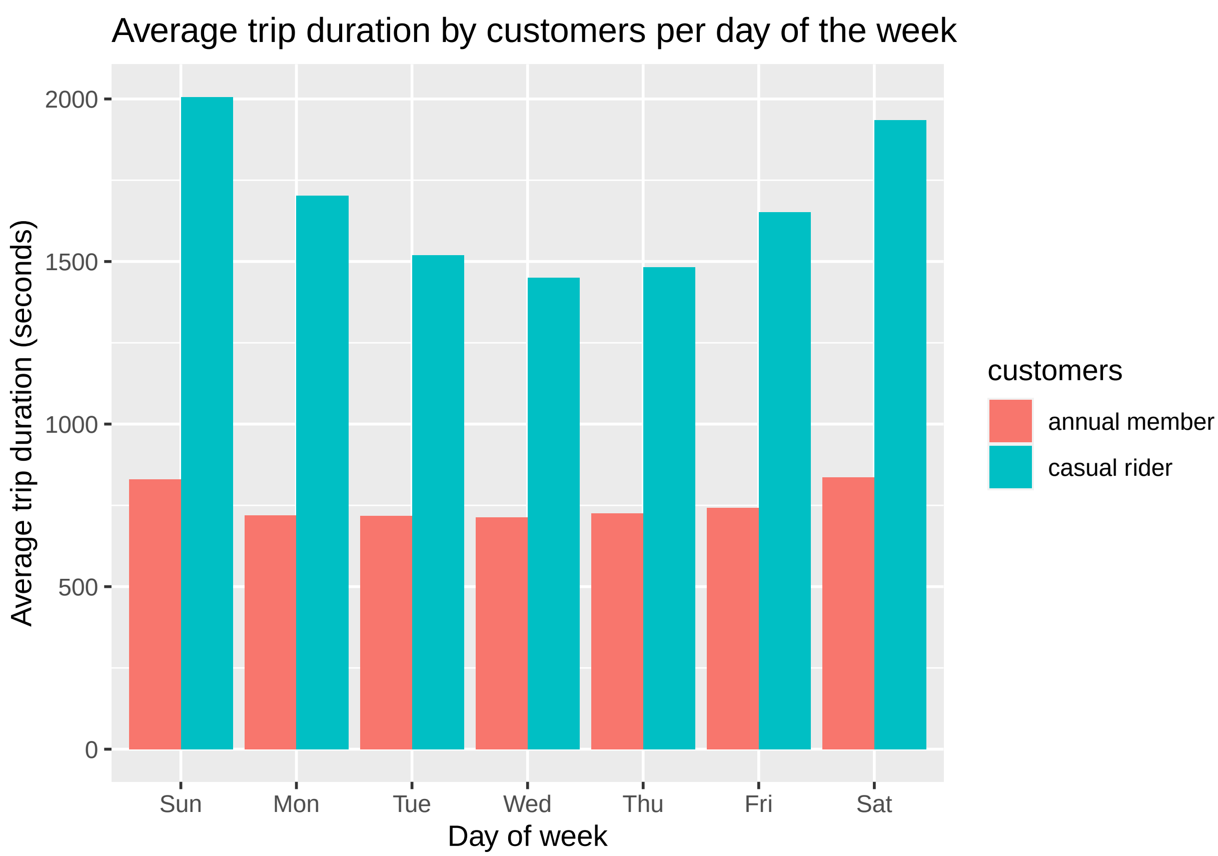
```
## `summarise()` has grouped output by 'customers'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   customers [2]
##    customers      day   number_of_trips average_trip_time
##    <chr>          <ord>           <int> <drtn>
##  1 annual member Sun            402040  829.4792 secs
##  2 annual member Mon            484532  718.7297 secs
##  3 annual member Tue            544350  717.1213 secs
##  4 annual member Wed            556882  713.5190 secs
##  5 annual member Thu            560837  726.1044 secs
##  6 annual member Fri            497434  741.9469 secs
##  7 annual member Sat            454394  835.9623 secs
##  8 casual rider  Sun            388779 2006.4384 secs
##  9 casual rider  Mon            275726 1702.2405 secs
```

```
## 10 casual rider  Tue           272624 1520.2851 secs
## 11 casual rider  Wed           284556 1450.4789 secs
## 12 casual rider  Thu           318437 1482.1343 secs
## 13 casual rider  Fri           350053 1651.2409 secs
## 14 casual rider  Sat           467873 1934.8136 secs
```

```
ggplot(tripdata_last_12_months %>% group_by(customers, day) %>%
    summarise(number_of_trips = n(),
    average_trip_time = mean(total_ride_duration))) +
  geom_col(position="dodge",
  mapping= aes(x = day, y = average_trip_time, fill = customers)) +
  labs(title="Average trip duration by customers per day of the week",
       x = "Day of week", y = "Average trip duration (seconds)")
```

```
## `summarise()` has grouped output by 'customers'. You can override using the
## `.groups` argument.
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```

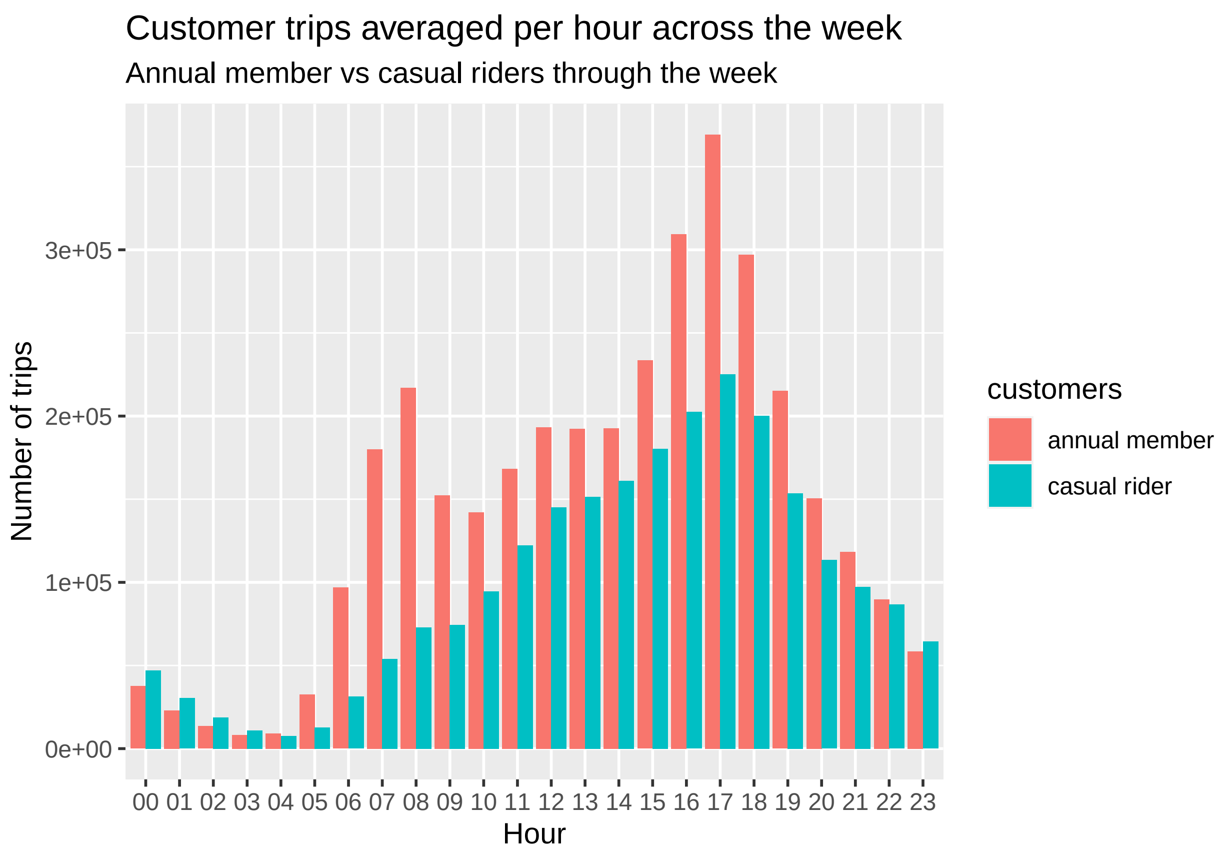


To further evidence that the annual members use the bikes to get to and from their places of employment, we can see a substantial peak from 7 - 9 AM and a greater one 4 - 6 PM. That structure is missing for the casual riders who only have one peak around 4 - 6 PM.

```
ggplot(tripdata_last_12_months %>%
         filter(day %in% c("Mon","Tue","Wed","Thu","Fri","Sat","Sun")) %>%
```

```
        group_by(customers, hour) %>%
        summarise(number_of_rides = n())) +
  geom_col(position="dodge",
          mapping= aes(x = hour, y = number_of_rides, fill = customers)) +
  labs(title="Customer trips averaged per hour across the week",
       subtitle = "Annual member vs casual riders through the week",
       x = "Hour", y = "Number of trips")
```

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

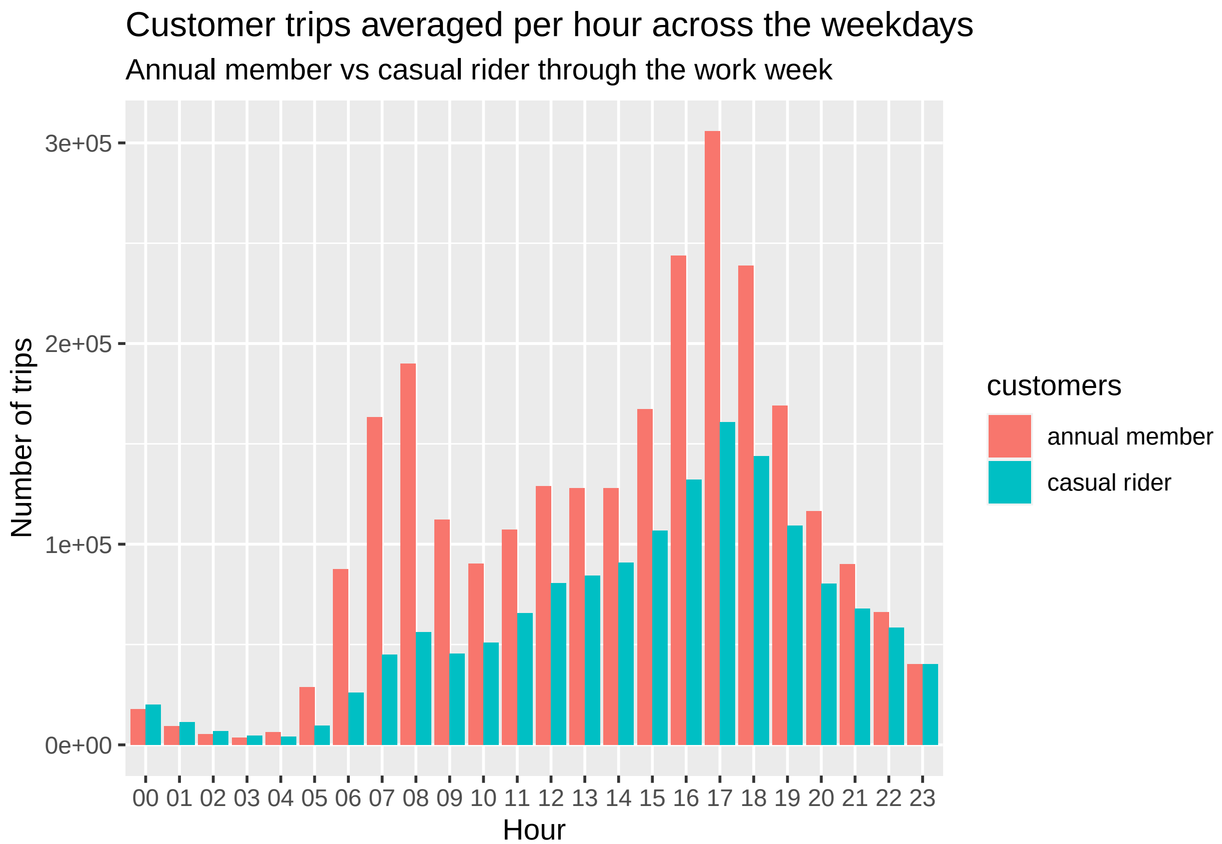


To test that hypothesis, lets narrow our search to only weekdays. It seems to hold true under these assumptions.

```
ggplot(tripdata_last_12_months %>%
        filter(day %in% c("Mon","Tue","Wed","Thu","Fri")) %>%
        group_by(customers, hour) %>%
        summarise(number_of_rides = n())) +
  geom_col(position="dodge",
          mapping= aes(x = hour, y = number_of_rides, fill = customers)) +
  labs(title="Customer trips averaged per hour across the weekdays",
       subtitle = "Annual member vs casual rider through the work week",
       x = "Hour", y = "Number of trips")
```

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

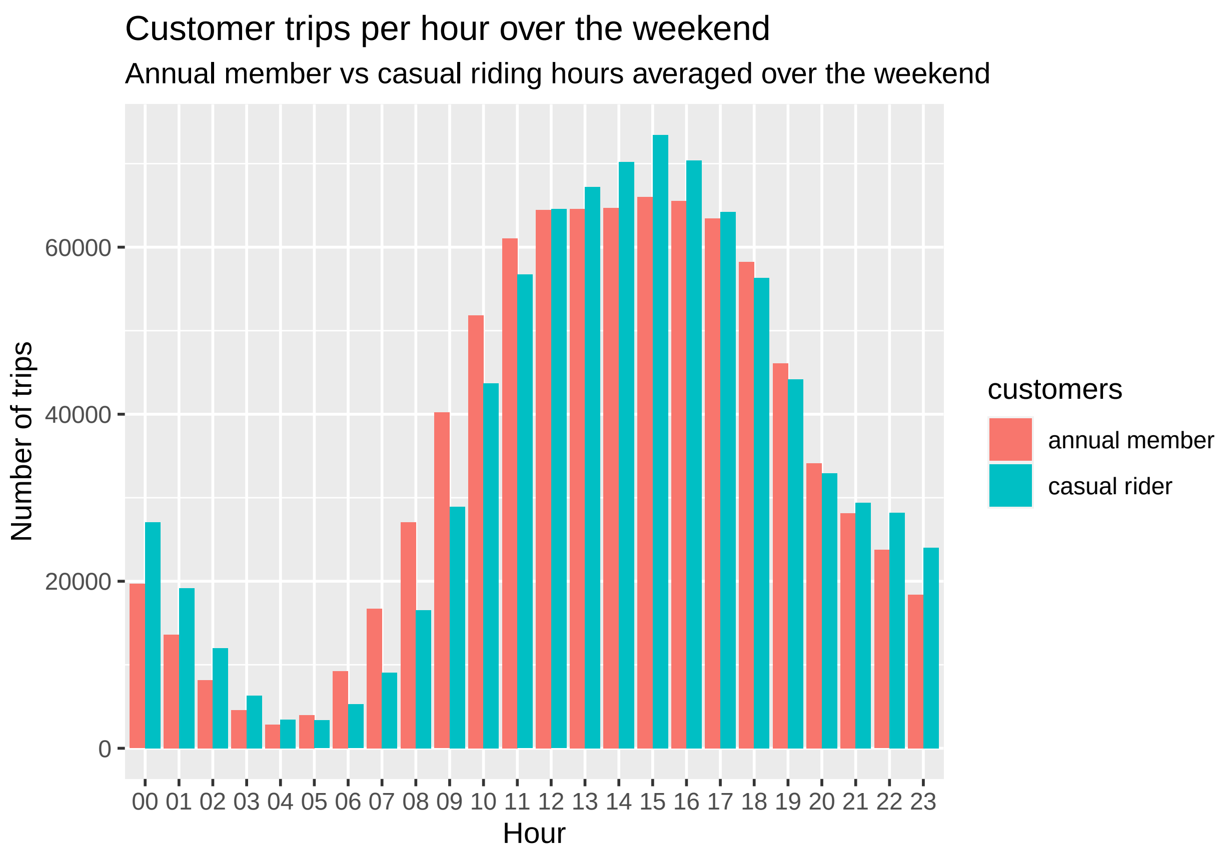
```
## `.groups` argument.
```



For good measure, lets see how the weekend compares by itself. It seems the annual members also do some weekend biking as well.

```
ggplot(tripdata_last_12_months %>%
         filter(day %in% c("Sat","Sun")) %>%
         group_by(customers, hour) %>%
         summarise(number_of_rides = n())) +
  geom_col(position="dodge",
           mapping= aes(x = hour, y = number_of_rides, fill = customers)) +
  labs(title="Customer trips per hour over the weekend",
    subtitle = "Annual member vs casual riding hours averaged over the weekend",
    x = "Hour", y = "Number of trips")
```

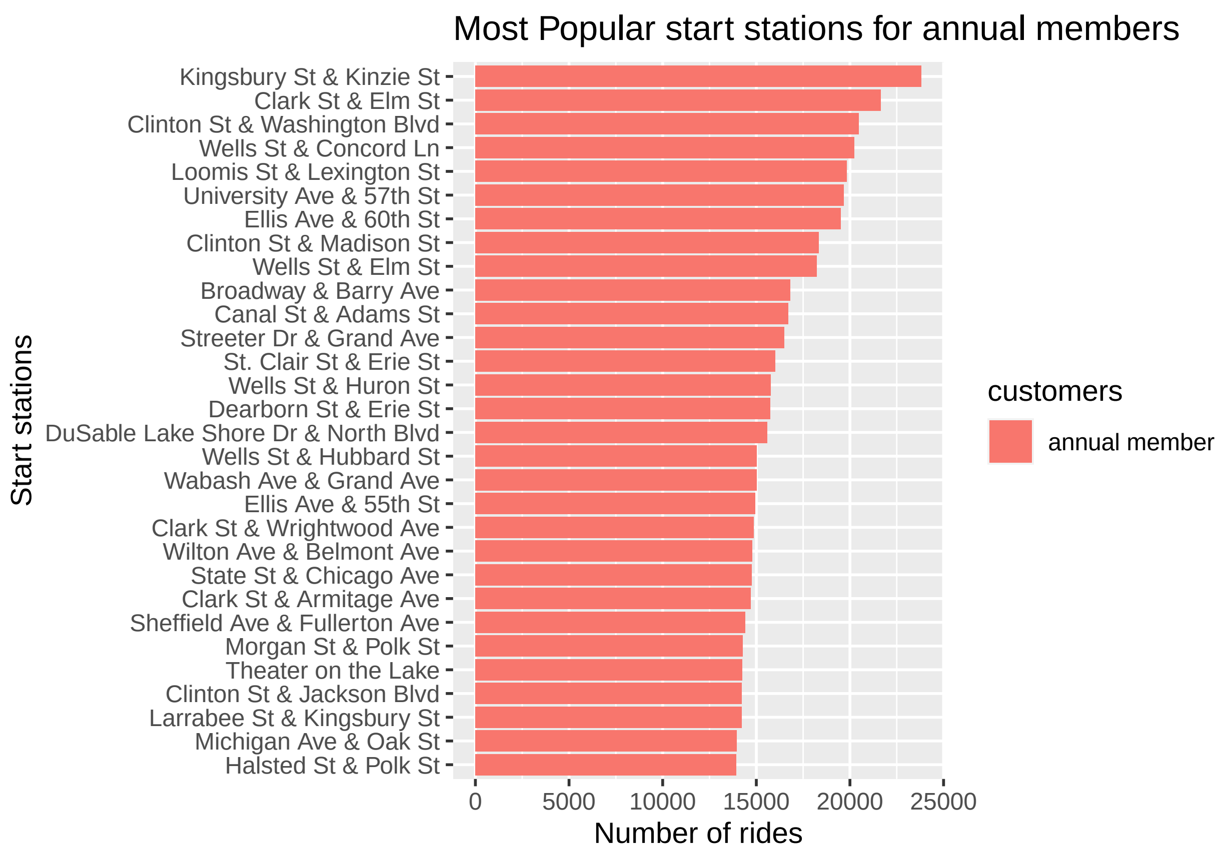
```
## `summarise()` has grouped output by 'customers'. You can override using the
## `.groups` argument.
```



Knowing the most utilized start stations will help our understanding of the customers.

```
ggplot(tripdata_last_12_months %>% drop_na() %>%
         filter(customers %in% "annual member") %>%
         group_by(start_station_name, customers) %>%
         summarize(count=n()) %>% arrange(desc(count)) %>%
         head(30)) +
  geom_col(mapping = aes(x = count, y= reorder(start_station_name, count),
                         fill= customers)) +
  labs(title="Most Popular start stations for annual members",
       x="Number of rides", y="Start stations")
```

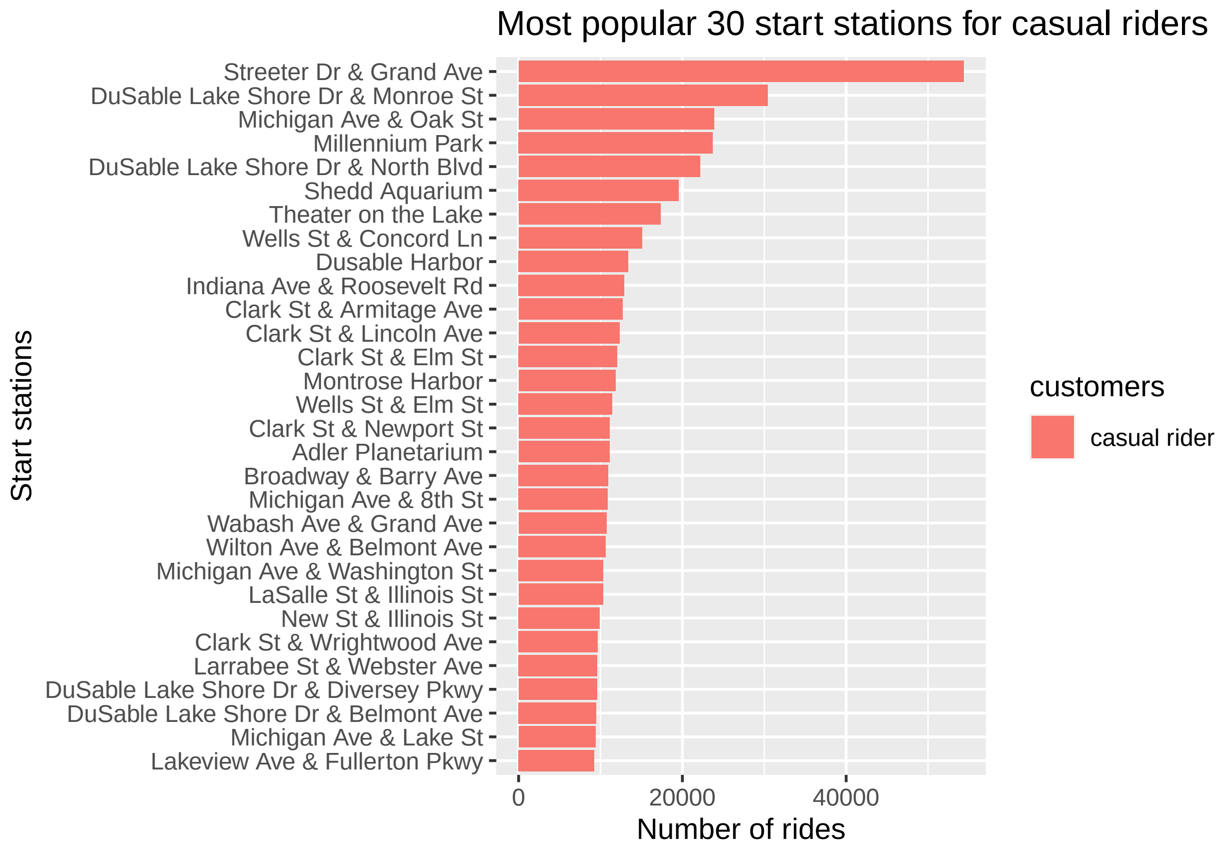
```
## `summarise()` has grouped output by 'start_station_name'. You can override
## using the `.groups` argument.
```



From these graphics, we know that Kingsburg Street & Kinzie St is the most popular station to depart from for members while Streeter Drive & Grand Avenue is the twice as utilized as the next popular station by the casual riders. There is a lot of overlap in these stations, 12 are shared.

```
ggplot(tripdata_last_12_months %>% drop_na() %>%
         filter(customers %in% "casual rider") %>%
         group_by(start_station_name, customers) %>%
         summarize(count=n()) %>%
         arrange(desc(count)) %>%
         head(30)) +
  geom_col(mapping = aes(x = count, y= reorder(start_station_name, count),
                       fill= customers)) +
  labs(title="Most popular 30 start stations for casual riders",
       x="Number of rides", y="Start stations")
```

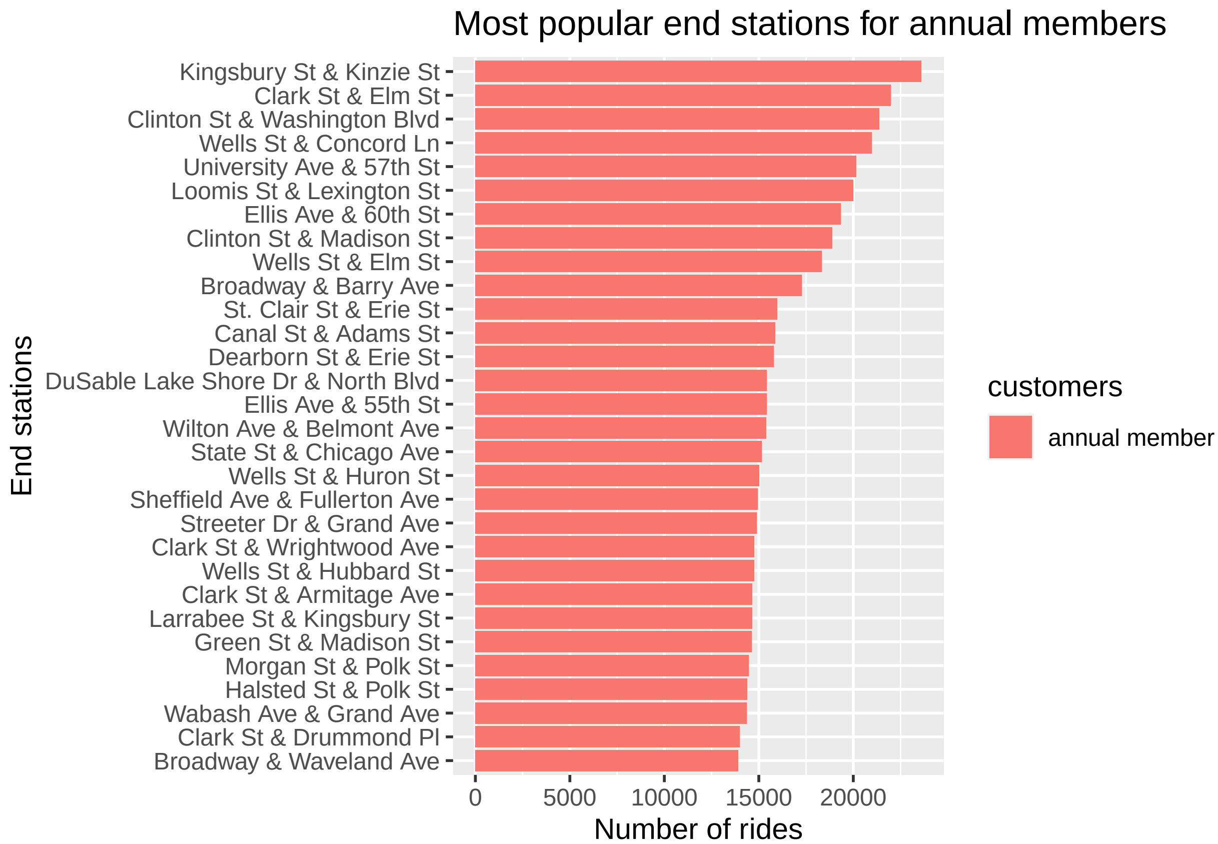
```
## `summarise()` has grouped output by 'start_station_name'. You can override
## using the `.groups` argument.
```



Let's see if the end stations have any patterns.

```
ggplot(tripdata_last_12_months %>% drop_na() %>%
         filter(customers %in% "annual member") %>%
         group_by(end_station_name, customers) %>%
         summarize(count=n()) %>% arrange(desc(count)) %>%
         head(30)) +
  geom_col(mapping = aes(x = count, y= reorder(end_station_name, count),
                         fill= customers)) +
  labs(title="Most popular end stations for annual members",
       x="Number of rides", y="End stations")
```

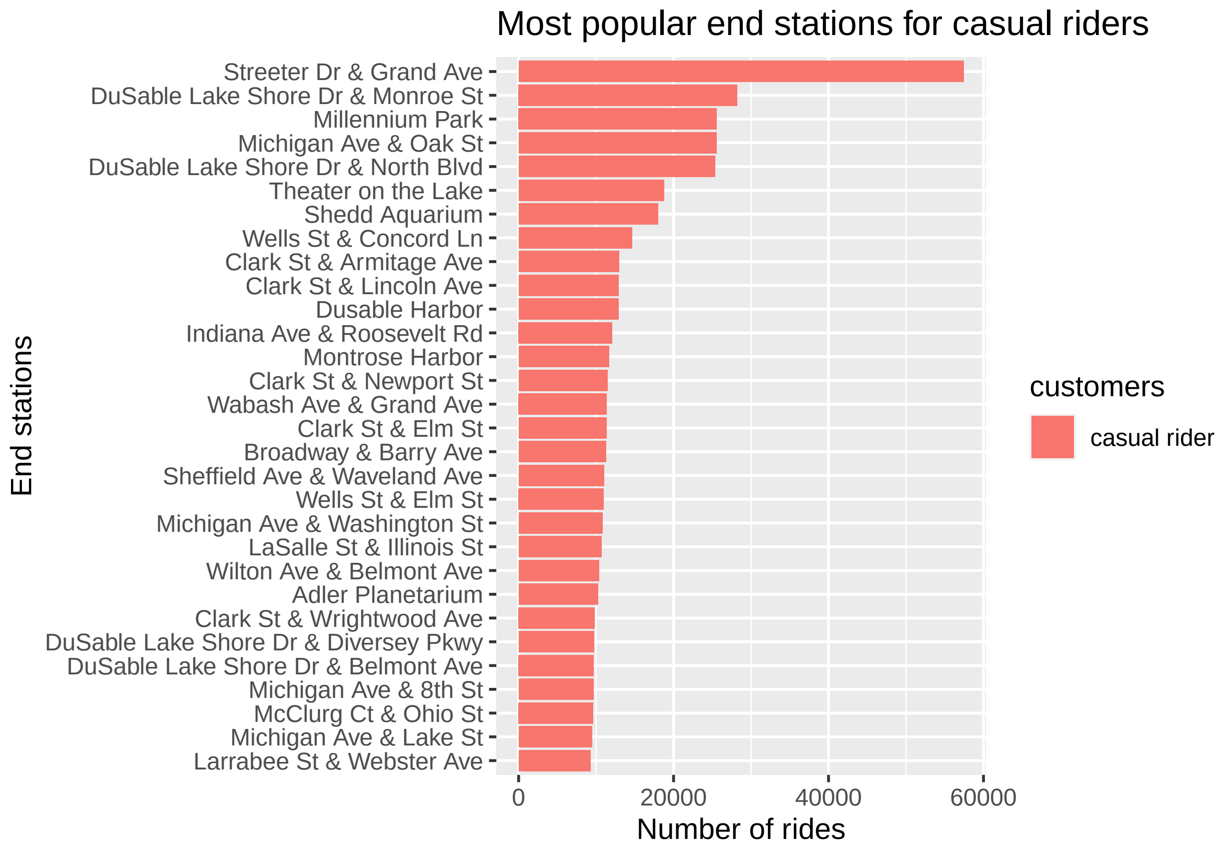
```
## `summarise()` has grouped output by 'end_station_name'. You can override using
## the `.groups` argument.
```



There is quite a bit of overlap between the most popular start and end stations relative to the customer type.

```
ggplot(tripdata_last_12_months %>%
         drop_na() %>%
         filter(customers %in% "casual rider") %>%
         group_by(end_station_name, customers) %>%
         summarize(count=n()) %>%
         arrange(desc(count)) %>%
         head(30)) +
  geom_col(mapping = aes(x = count, y= reorder(end_station_name, count),
                         fill= customers)) +
  labs(title="Most popular end stations for casual riders",
       x="Number of rides", y="End stations")
```

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

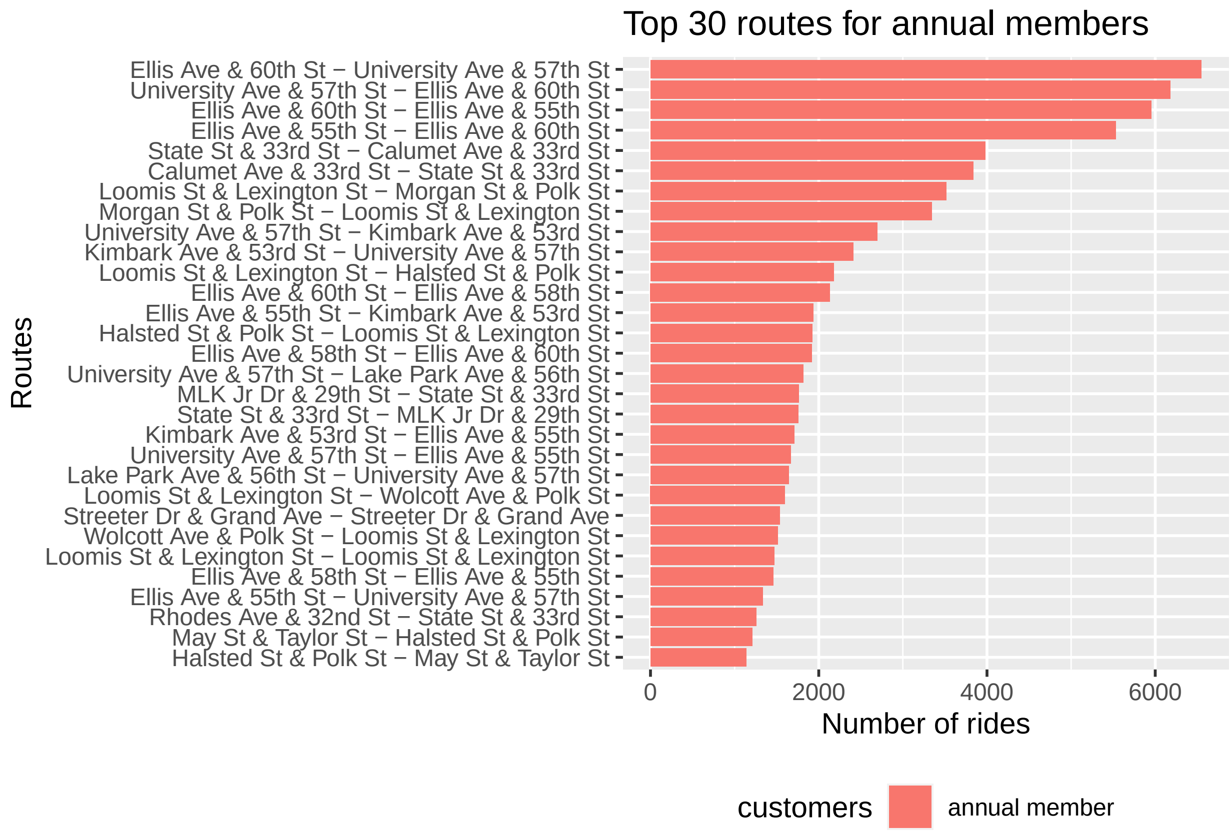


It would behoove us to locate the most popular routes customers are taking, so we'll add a column tracking that.

```
tripdata_last_12_months$route <-
  paste(tripdata_last_12_months$start_station_name, "-",
        tripdata_last_12_months$end_station_name)
```

```
ggplot(tripdata_last_12_months %>%
         drop_na() %>%
         filter(customers %in% "annual member") %>%
         group_by(route, customers) %>%
         summarize(count=n()) %>%
         arrange(-count) %>%
         head(30)) +
  geom_col(position="dodge", mapping = aes(x = count, y= reorder(route, count),
                                          fill= customers)) +
  theme(legend.position = "bottom") +
  labs(title="Top 30 routes for annual members",
       x="Number of rides", y="Routes")
```

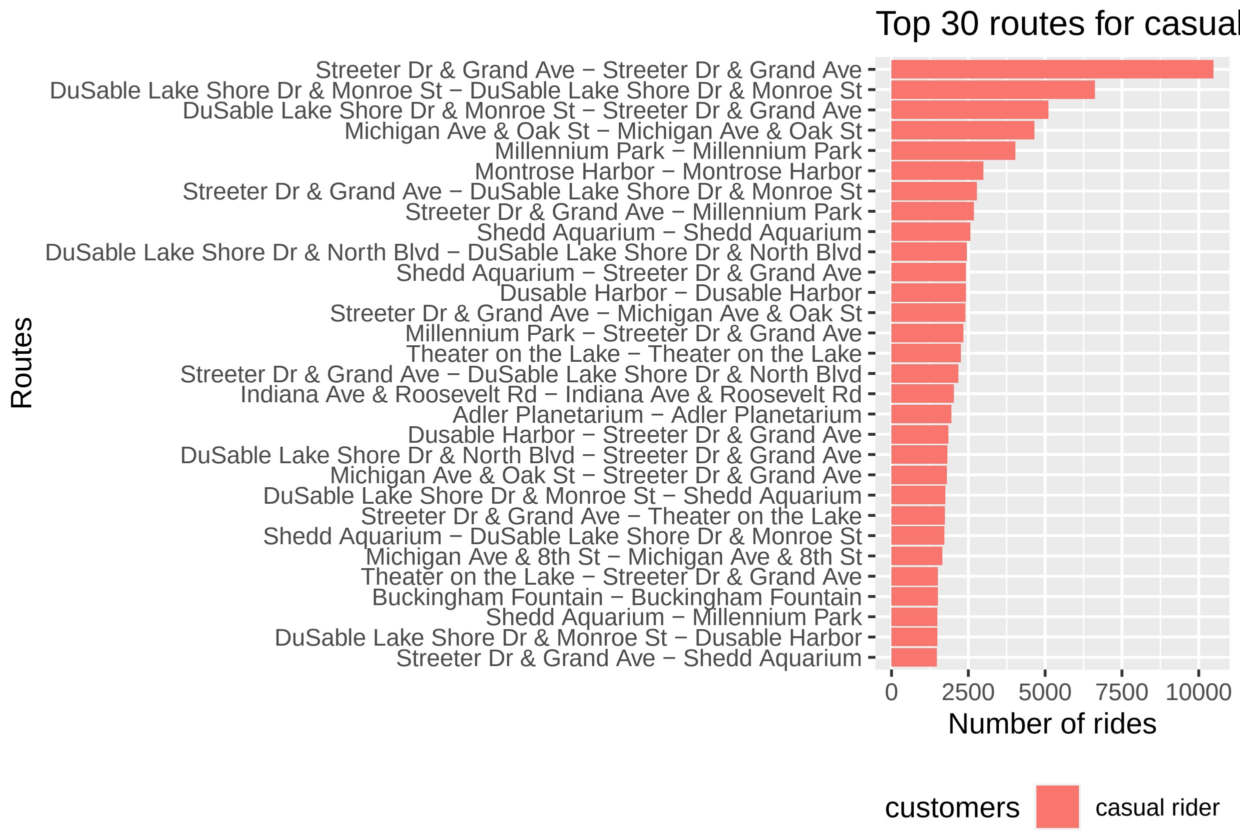
```
## `summarise()` has grouped output by 'route'. You can override using the
## `.groups` argument.
```



Here we can see the most popular routes for both annual members and casual riders. They only share one route, that being the Streeter Drive & Grand Avenue to Streeter Drive & Grand Avenue.

```
ggplot(tripdata_last_12_months %>%
         drop_na() %>%
         filter(customers %in% "casual rider") %>%
         group_by(route, customers) %>%
         summarize(count=n()) %>%
         arrange(-count) %>%
         head(30)) + geom_col(position="dodge", mapping = aes(x = count,
                           y= reorder(route, count), fill= customers)) +
  theme(legend.position = "bottom") +
  labs(title="Top 30 routes for casual riders", x="Number of rides", y="Routes")
```

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



Findings

My analysis turned up the following points from the data:

- 60% of the total trips during this time range were done by annual members, the remaining 40% by casual riders.
- Electric bikes are more popular than their classic counter parts for casuals, but roughly equal in popularity for members. A side note, only the casuals used the docked bikes and then only a minority of them.
- Winter months (November to February) see a major drop in usage for both customer types and summer months (May to September) feature a spike in the bikes' use, almost to parity for June and July. The winter months' effect is particularly pronounced for the casual riders. Annual member use peaks in August, as compared to July for casuals.
- The distribution of the hourly data for annual member trips implies they generally ride to and from work on the weekdays, a fact which hints that they are likely to be Chicago residents.
- Annual members started the most trips of any station at Kingsbury Street & Kinzie Street while the casual riders began more trips at Streeter Drive & Grand Avenue.
- The top 4 routes for annual members and casual riders are a large chunk of the total for each, particularly casual rider's top route. The higher average trip duration for members implies they travel farther than their casual counterparts.

Recommendations

Based on this analysis, I can think of a few improvements to get more out of future endeavors:

- Deeper customer data collection would yield more potential insights, collect characteristics such as age, gender, residence, and other relevant factors to better understanding of their needs.
- Ensure the incoming data has all of the variables, such as start and end stations being present, proper time accounting and station IDs.
- Identify residents who are casual riders to launch a better targeted advertisement campaign.