Cyclistic Bike-Sharing Data Explorations

Google Data Analytics Capstone (case study)

Paw Hermansen

2022-09-03

Scenario and the Business Task

Cyclistic is a bike-share company in Chicago. Cyclistic has two kinds of customers, casual riders who pay per ride or per day, and members who pay annually.

Cyclistic wants to convert more casual riders into annual members and as part of this they need to know how annual members and casual riders use Cyclistic bikes differently.

The Business Task is:

> Analyze the available dataset to answer how annual members and casual riders use Cyclistic bikes differently.

Getting the Data

The data is downloaded from https://divvy-tripdata.s3.amazonaws.com/index.html and has the form of 12 zipped csv files—one file for each month of the period August 2021 to July 2022, both included.

The data is public and made available by Motivate International Inc. under this license. Any information about riders has been removed. This means that this data can be used without having concerns about data privacy but it also means that we cannot include for example the riders sex, age, place of residence, number of members, and number of unique casual riders in the analysis which surely would have been a big source of very useful information.

The data seems trustworthy. As far as I can assess the data is original, data is available for a continues period over several years including very current data and I believe it contains all the rides. Also, it is linked-to by the Google Data Analytics Certification Capstone assignment.

Because the data contains the full set of rides in the examined period for all riders and all geographies, or in short the full population, it has no problems with selection bias—except from the possibility that missing values are skew, for example if casual riders have missing many more values than members. I will look at missing values and the possibility of skewness in missing values later in this report.

```
# Unzip the zipped files
all_zipfilenames <- list.files("data", pattern="\\d{6}-divvy-tripdata\\.zip$", ignore.case=TRUE, full.na
sapply(all_zipfilenames, unzip, exdir="data", overwrite=TRUE)
```

Counting the number of rows in the csv files show that it's close to six million rows in total. Because of this I decide to do the analysis in R.

```
# List the names of the csv files
all_csvfilenames <- list.files("data", pattern="\\d{6}-divvy-tripdata\\.csv$", ignore.case=TRUE, full.na
```

```
# Find the number of rows in each csv file
number_of_csvfile_rows <- list()
total_num_rows <- 0
for (filename in all_csvfilenames) {
  num_rows <- peek_count_lines(filename)
  total_num_rows <- total_num_rows + num_rows
  cat(sprintf("%s, number of rows: %6d\n", filename, num_rows))
  number_of_csvfile_rows <- append(number_of_csvfile_rows, num_rows)
}
```

```
## data/202108-divvy-tripdata.csv, number of rows: 804353
## data/202109-divvy-tripdata.csv, number of rows: 756148
## data/202110-divvy-tripdata.csv, number of rows: 631227
## data/202111-divvy-tripdata.csv, number of rows: 359979
## data/202112-divvy-tripdata.csv, number of rows: 247541
## data/202201-divvy-tripdata.csv, number of rows: 103771
## data/202202-divvy-tripdata.csv, number of rows: 115610
## data/202203-divvy-tripdata.csv, number of rows: 284043
## data/202204-divvy-tripdata.csv, number of rows: 371250
## data/202205-divvy-tripdata.csv, number of rows: 634859
## data/202206-divvy-tripdata.csv, number of rows: 769205
## data/202207-divvy-tripdata.csv, number of rows: 823489
```

```
cat("Total number of rows = ", total_num_rows, "\n")
```

```
## Total number of rows =  5901475
```

Visually inspecting the top lines of two of the files show that they both have a header as the first row and have the data start in the second row.

```
peek_head("data/202108-divvy-tripdata.csv", n = 3)
```

```
## ride_id,rideable_type,started_at,ended_at,start_station_name,start_station_id,end_station_name,end_s
## 99103BB87CC6C1BB,electric_bike,2021-08-10 17:15:49,2021-08-10 17:22:44,,,,,41.77,-87.68,41.77,-87.68
## EAFCCCFB0A3FC5A1,electric_bike,2021-08-10 17:23:14,2021-08-10 17:39:24,,,,,41.77,-87.68,41.77,-87.63
```

```
peek_head("data/202207-divvy-tripdata.csv", n = 3)
```

```
## ride_id,rideable_type,started_at,ended_at,start_station_name,start_station_id,end_station_name,end_s
## 954144C2F67B1932,classic_bike,2022-07-05 08:12:47,2022-07-05 08:24:32,Ashland Ave & Blackhawk St,132
## 292E027607D218B6,classic_bike,2022-07-26 12:53:38,2022-07-26 12:55:31,Buckingham Fountain (Temp),155
```

The following check confirms that all the files have a header row and that the header is the same for all the files. This means that I expect the csv files to have columns of the same type of data in the same order. It is also noted that some of the data values are missing (when there is nothing between the commas).

```
header_rows <- sapply(all_csvfilenames, readLines, n=1)
cat("Number of different header rows =", length(unique(header_rows)))
```

```
## Number of different header rows = 1
```

Finally the csv files are read and combined into a data frame named `tripdata_df`, or more exactly into a R tibble but I'll just call it a data frame in the rest of this report.

```
tripdata_df <- do.call(rbind, lapply(all_csvfilenames, read_csv, col_types=cols()))

glimpse(tripdata_df)
```

```
## Rows: 5,901,463
```

```
## Columns: 13
## $ ride_id            <chr> "99103BB87CC6C1BB", "EAFCCCFB0A3FC5A1", "9EF4F46C57~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2021-08-10 17:15:49, 2021-08-10 17:23:14, 2021-08-~
## $ ended_at           <dttm> 2021-08-10 17:22:44, 2021-08-10 17:39:24, 2021-08-~
## $ start_station_name <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_station_id   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, "Clark St & Grace St", ~
## $ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, "TA1307000127", NA, NA,~
## $ start_lat          <dbl> 41.77000, 41.77000, 41.95000, 41.97000, 41.79000, 4~
## $ start_lng          <dbl> -87.68000, -87.68000, -87.65000, -87.67000, -87.600~
## $ end_lat            <dbl> 41.77000, 41.77000, 41.97000, 41.95000, 41.77000, 4~
## $ end_lng            <dbl> -87.68000, -87.63000, -87.66000, -87.65000, -87.620~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

In all we have 5901463 rows for the twelve months.

Data Integrity and Data Cleaning

Check of Data Loading

To be sure we have loaded the data correctly I find the number of rides in each month of the loaded data and compare them with the number of rows in the csv files found earlier. It is seen that every csv file has exactly one row extra which is the header row present in all the csv files. This means that all rows of the csv files have been loaded.

```
tripdata_df %>%
  mutate(date_yearmon = format(started_at, "%Y-%m")) %>%
  group_by(date_yearmon) %>%
  summarise(number_of_rows_in_df = n()) %>%
  add_column(number_of_rows_in_csvfiles = unlist(number_of_csvfile_rows)) %>%
  mutate(difference = number_of_rows_in_csvfiles - number_of_rows_in_df)
```

```
## # A tibble: 12 x 4
##    date_yearmon number_of_rows_in_df number_of_rows_in_csvfiles difference
##    <chr>                       <int>                       <int>      <int>
##  1 2021-08                    804352                      804353          1
##  2 2021-09                    756147                      756148          1
##  3 2021-10                    631226                      631227          1
##  4 2021-11                    359978                      359979          1
##  5 2021-12                    247540                      247541          1
##  6 2022-01                    103770                      103771          1
##  7 2022-02                    115609                      115610          1
##  8 2022-03                    284042                      284043          1
##  9 2022-04                    371249                      371250          1
## 10 2022-05                    634858                      634859          1
## 11 2022-06                    769204                      769205          1
## 12 2022-07                    823488                      823489          1
```

ride_id

A unique key of R type Character (text string) for each ride. All rows have a different `ride_id` and they are all 16 characters long.

```
cat("Number of missing values: ", sum(is.na(tripdata_df$ride_id) | tripdata_df$ride_id == ""), "\n")
```

```
## Number of missing values:  0
```

```
cat("Number of duplets: ", nrow(tripdata_df) - n_distinct(tripdata_df$ride_id), "\n")
```

```
## Number of duplets:  0
```

```
tripdata_df %>% group_by(str_length(ride_id)) %>% summarize(count = n())
```

```
## # A tibble: 1 x 2
##   `str_length(ride_id)`   count
##                   <int>   <int>
## 1                    16 5901463
```

rideable_type

Three different types of bikes are present: `classic_bike`, `docked_bike`, `electric_bike`. No row is missing this value (or it would have shown-up as a fourth value in the following code-chunk and the three counts would not have summed up to the total number of rows).

Because this is categorial data, i.e. only a few fixed different values exist, I change its type in `tripdata_df` to the R type factor.

```
table(tripdata_df$rideable_type)
```

```
##
##  classic_bike   docked_bike electric_bike
##       3055641        226728       2619094
```

```
tripdata_df$rideable_type <- as.factor(tripdata_df$rideable_type)
str(tripdata_df$rideable_type)
```

```
##  Factor w/ 3 levels "classic_bike",..: 3 3 3 3 3 3 3 3 3 3 ...
```

started_at, ended_at

Start-time and end-time of each ride of type date-time. All rows have both `started_at` and `ended_at`.

```
cat("Number of missing started_at: ", sum(is.na(tripdata_df$started_at)), "\n")
```

```
## Number of missing started_at:  0
```

```
cat("Number of missing ended_at:   ", sum(is.na(tripdata_df$ended_at)), "\n")
```

```
## Number of missing ended_at:    0
```

start_station_name, end_station_name, start_station_id, end_station_id

Because about 15% of the station names and station ids are missing I decide to not use the station data and I delete the columns.

```
num_rows = nrow(tripdata_df)
missing_ssn <- sum(is.na(tripdata_df$start_station_name) | tripdata_df$start_station_name == "")
missing_esn <- sum(is.na(tripdata_df$end_station_name) | tripdata_df$end_station_name == "")
missing_ssi <- sum(is.na(tripdata_df$start_station_id) | tripdata_df$start_station_id == "")
missing_esi <- sum(is.na(tripdata_df$end_station_id) | tripdata_df$end_station_id == "")

cat(sprintf("Number of missing start_station_names: %6d = %.1f%%\n", missing_ssn, 100 * missing_ssn / nu
```

```
## Number of missing start_station_names: 860786 = 14.6%
```

```
cat(sprintf("Number of missing end_station_names:   %6d = %.1f%%\n", missing_esn, 100 * missing_esn / nu
```

```
## Number of missing end_station_names:   919896 = 15.6%
```

```
cat(sprintf("Number of missing start_station_ids:   %6d = %.1f%%\n", missing_ssi, 100 * missing_ssi / nu
```

```
## Number of missing start_station_ids:   860784 = 14.6%
```

```
cat(sprintf("Number of missing end_station_ids:     %6d = %.1f%%\n", missing_esi, 100 * missing_esi / nu
```

```
## Number of missing end_station_ids:     919896 = 15.6%
```

```
tripdata_df <- select(tripdata_df, -c(start_station_name, end_station_name, start_station_id, end_statio
glimpse(tripdata_df)
```

```
## Rows: 5,901,463
## Columns: 9
## $ ride_id       <chr> "99103BB87CC6C1BB", "EAFCCCFB0A3FC5A1", "9EF4F46C57AD234~
## $ rideable_type <fct> electric_bike, electric_bike, electric_bike, electric_bi~
## $ started_at    <dttm> 2021-08-10 17:15:49, 2021-08-10 17:23:14, 2021-08-21 02~
## $ ended_at      <dttm> 2021-08-10 17:22:44, 2021-08-10 17:39:24, 2021-08-21 02~
## $ start_lat     <dbl> 41.77000, 41.77000, 41.95000, 41.97000, 41.79000, 41.810~
## $ start_lng     <dbl> -87.68000, -87.68000, -87.65000, -87.67000, -87.60000, -~
## $ end_lat       <dbl> 41.77000, 41.77000, 41.97000, 41.95000, 41.77000, 41.800~
## $ end_lng       <dbl> -87.68000, -87.63000, -87.66000, -87.65000, -87.62000, -~
## $ member_casual <chr> "member", "member", "member", "member", "member", "membe~
```

start_lat, start_lng

The geographical latitude and longitude of type R `double`, i.e. a number, of the trip start point. No values are missing.

```
cat("Number of missing start_lat: ", sum(is.na(tripdata_df$start_lat)), "\n")
```

```
## Number of missing start_lat:  0
```

```
cat("Number of missing start_lng: ", sum(is.na(tripdata_df$start_lng)), "\n")
```

```
## Number of missing start_lng:  0
```

end_lat, end_lng

The geographical latitude and longitude of type R `double`, i.e. a number, of the trip end point. 5590 values are missing.

```
cat("Number of missing end_lat: ", sum(is.na(tripdata_df$end_lat)), "\n")
```

```
## Number of missing end_lat:  5590
```

```
cat("Number of missing end_lng: ", sum(is.na(tripdata_df$end_lng)), "\n")
```

```
## Number of missing end_lng:  5590
```

It turns out that the two values are missing in exactly the same rows so 5590 rows in all are affected. That is about 0.1% of all the rows. This number is so small that deleting the rows with missing values will not change the results of this analyses in any visible way. So I delete the involved 5590 rows.

If the missing values are in rows that for example are exactly of one kind of `member_casual` then deleting these rows will introduce skewness and bias but as shown the maximal possible bias is too small to influence

the results in this analyses.

```
cat("Number of at least one missing: ", sum(is.na(tripdata_df$end_lat) | is.na(tripdata_df$end_lng)), "
```

```
## Number of at least one missing:  5590
```

```
tripdata_df <- tripdata_df %>% filter(!is.na(tripdata_df$end_lat) & !is.na(tripdata_df$end_lng))
```

member_casual

Two different possibilities are present: `casual`, `member`. No row is missing this value (or it would have shown-up as a third value in the following code-chunk and the two counts would not have summed up to the total number of rows).

Because this is categorical data, i.e. only a few fixed different values exist, I change its type in `tripdata_df` to the R type factor.

```
levels(tripdata_df$member_casual)
```

```
## NULL
```

```
tripdata_df$member_casual <- as.factor(tripdata_df$member_casual)
str(tripdata_df$member_casual)
```

```
##  Factor w/ 2 levels "casual","member": 2 2 2 2 2 2 2 2 2 2 ...
```

Explorations Into the Differences Between Casual Riders and Members

Number of Rides by Casual Riders and Members

```
cat(sprintf("Number of Rides by Casual Riders: %d  %.0f%%\n", nrow(tripdata_df[tripdata_df$member_casual
```

```
## Number of Rides by Casual Riders: 2517435  43%
```

```
cat(sprintf("Number of Rides by Members      : %d  %.0f%%\n", nrow(tripdata_df[tripdata_df$member_casual
```

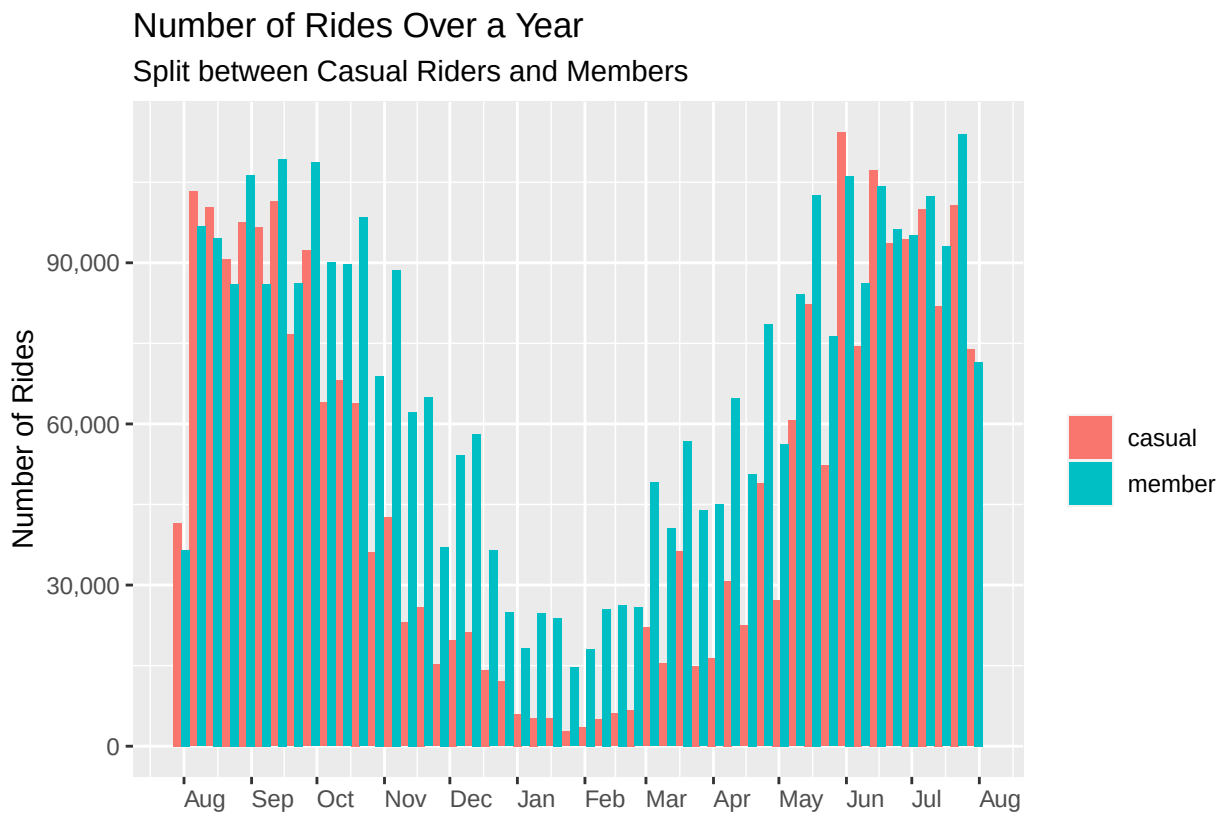
```
## Number of Rides by Members      : 3378438  57%
```

The data available doesn't give any information about the number of member or different casual riders and so I cannot make any conclusions about how often the two different rider types go for a ride.

Seasonal

During summer months approximately half of the riders are casual rides and half are members. Far fewer rides are done during the Chicago winter months and especially the casual riders are not riding during winter months. During January and February more than 80% of the rides are done by members.

```
tripdata_df %>%
  ggplot(aes(x=as.Date(started_at), fill=member_casual)) +
  geom_histogram(bins=50, position = "dodge") +
  scale_y_continuous(labels=scales::comma_format(big.mark=',', decimal.mark = '.')) +
  scale_x_date(date_breaks="1 month", date_labels="%b") +
  theme(axis.text.x=element_text(hjust = 0)) +
  labs(title="Number of Rides Over a Year", subtitle="Split between Casual Riders and Members", x="", y=
```




```
tripdata_df %>%
  filter(month(started_at) %in% c(1,2)) %>%
  group_by(member_casual) %>% summarize(number_of_rides_in_Jan_and_Feb=n()) %>% mutate(number_of_rides_
```

```
## # A tibble: 2 x 3
##   member_casual number_of_rides_in_Jan_and_Feb number_of_rides_percents
##   <fct>                                 <int>                    <dbl>
## 1 casual                                39824                       18
## 2 member                               179392                       82
```

```
tripdata_df %>%
  filter(month(started_at) %in% c(5,6,7,8)) %>%
  group_by(member_casual) %>% summarize(number_of_rides_in_May_to_Aug=n()) %>% mutate(number_of_rides_p
```

```
## # A tibble: 2 x 3
##   member_casual number_of_rides_in_May_to_Aug number_of_rides_percents
##   <fct>                                 <int>                    <dbl>
## 1 casual                              1465152                       48
## 2 member                              1563320                       52
```

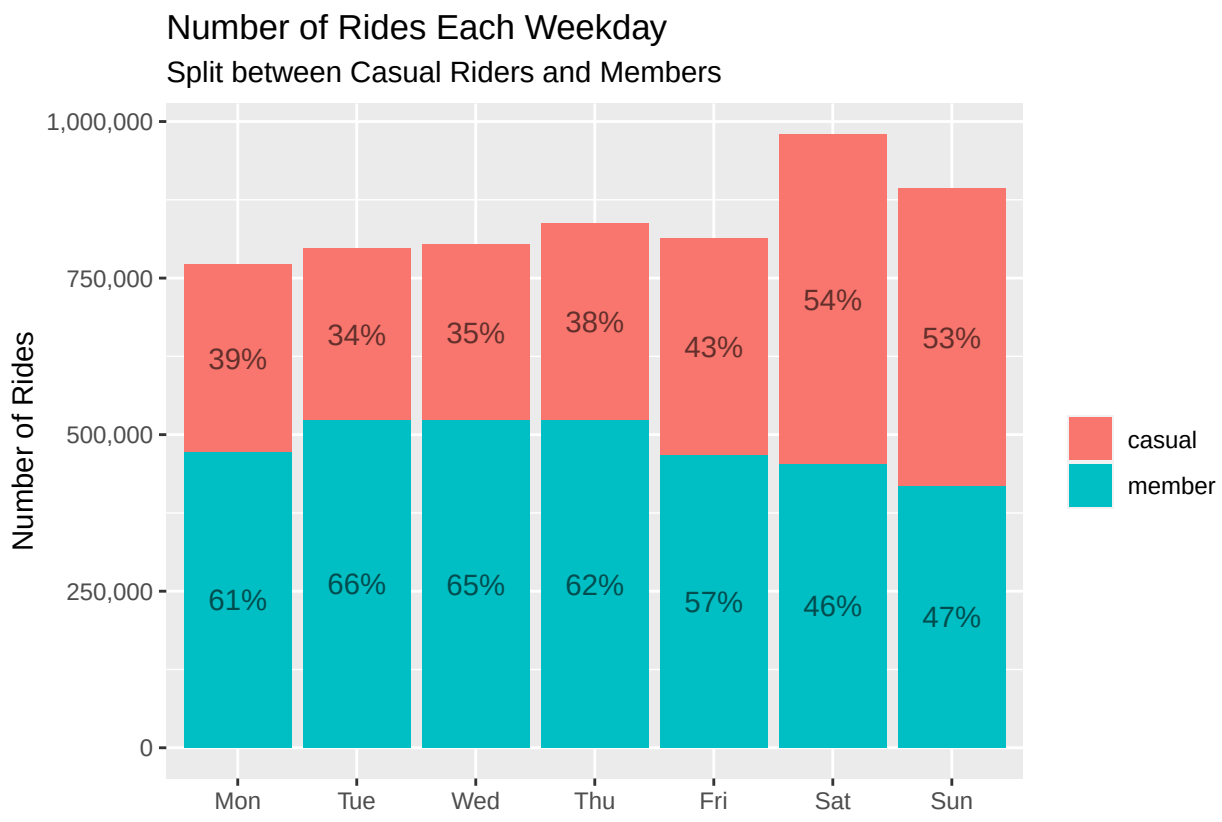
Days of the Week

During weekends more than half the rides are done by casual riders while the opposite is true during weekdays. This trend mostly happens because casual riders ride more during weekends than during the rest of the week while members ride more evenly over the full week, though with a very small tendency to ride less during weekends.

Because casual riders are so much more inactive during the winter months, this trend must be strongest

during the summer months.

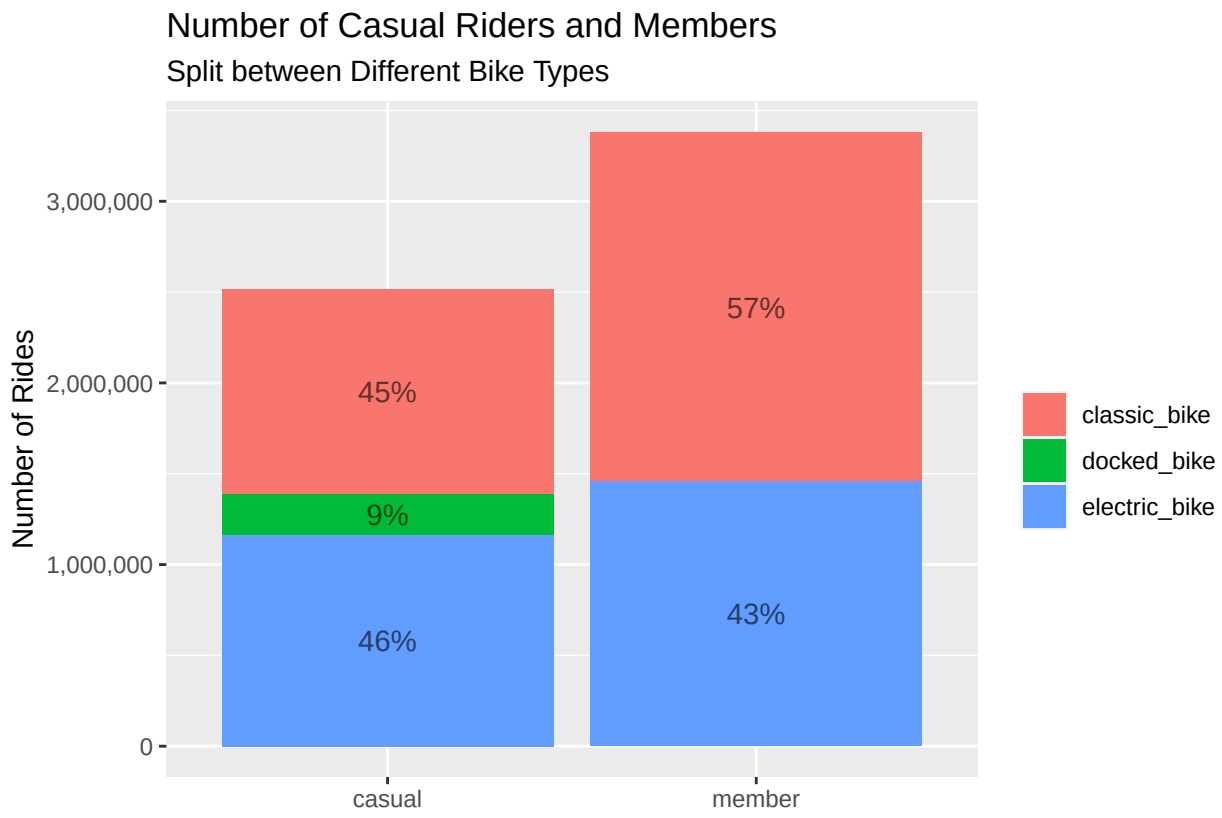
```
tripdata_df %>%
  group_by(weekday=wday(started_at, label=TRUE, week_start=1), member_casual) %>%
  summarize(number_of_rides=n()) %>%
  mutate(number_of_rides_prop = number_of_rides / sum(number_of_rides)) %>%
  ggplot(aes(x=weekday, y=number_of_rides, fill=member_casual)) + geom_bar(stat="identity") +
  geom_text(position=position_stack(vjust = 0.5), aes(label=scales::percent(number_of_rides_prop, accura
  scale_y_continuous(labels=scales::comma_format(big.mark=',', decimal.mark = '.')) +
  labs(title="Number of Rides Each Weekday", subtitle="Split between Casual Riders and Members", x="", y
```



Rideable Type

Members mostly ride a classical bike while casual riders are split evenly between clissical bikes and electrical bikes with a small group on docked bikes.

```
tripdata_df %>%
  group_by(member_casual, rideable_type) %>%
  summarize(number_of_rides=n()) %>%
  mutate(number_of_rides_prop = number_of_rides / sum(number_of_rides)) %>%
  ggplot(aes(x=member_casual, y=number_of_rides, fill=rideable_type)) + geom_bar(stat="identity") +
  geom_text(position=position_stack(vjust = 0.5), aes(label=scales::percent(number_of_rides_prop, accura
  scale_y_continuous(labels=scales::comma_format(big.mark=',', decimal.mark = '.')) +
  labs(title="Number of Casual Riders and Members", subtitle="Split between Different Bike Types", x=""
```

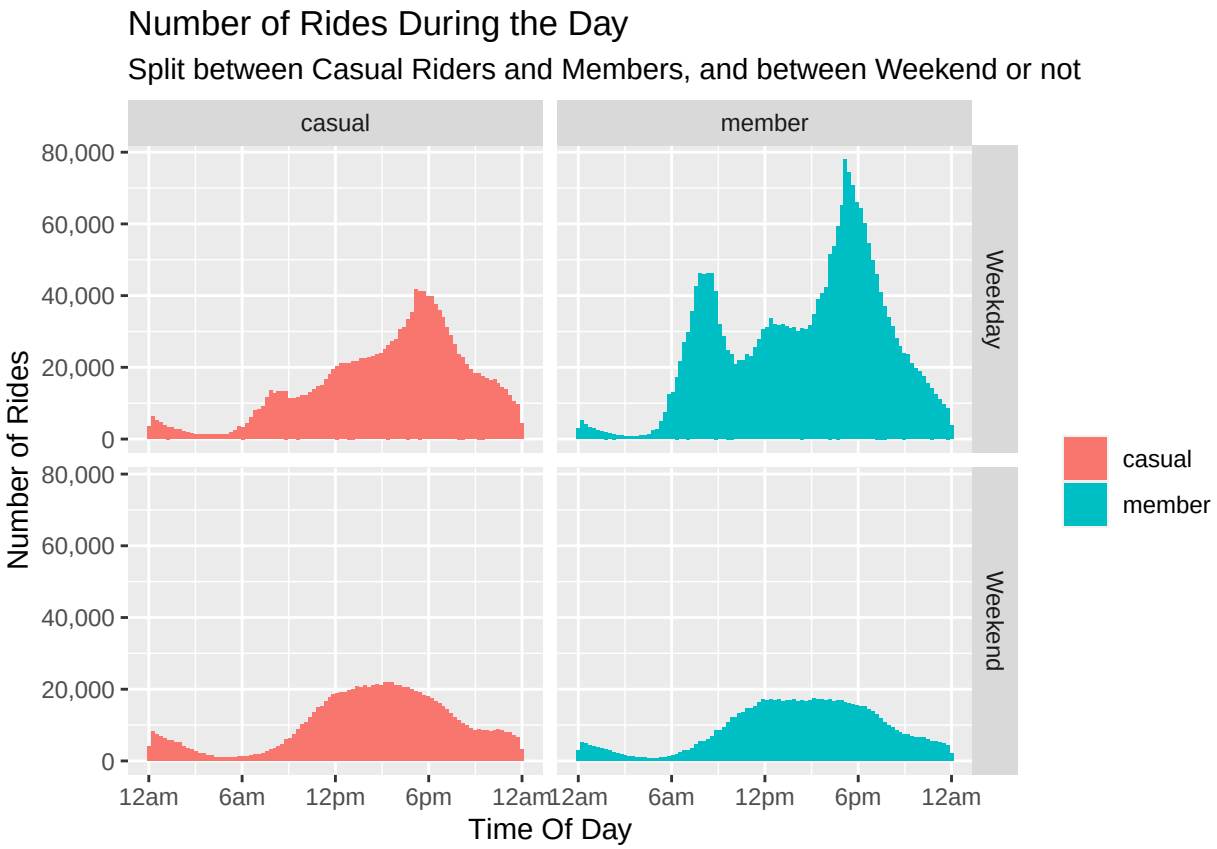


Time of Day

Members have a very clear pattern showing many rides in the hours just before and after work-hours on weekdays, Monday to Friday. Casual riders show a noticeable smaller peak right after work-hours and only a nearly invisible peak right before work-hours

During weekends neither casual riders nor members show such peaks.

```
tripdata_df %>%
  mutate(part_of_week = case_when(wday(started_at) %in% c(1,7) ~ "Weekend", TRUE ~ "Weekday")) %>%
  mutate(time_of_day = 3600*hour(started_at) + 60*minute(started_at) + second(started_at)) %>%
  ggplot(aes(x=time_of_day)) + geom_histogram(bins=96) +
  facet_grid(part_of_week ~ member_casual) + aes(fill=member_casual) +
  scale_x_continuous(breaks = c(0, 6*3600, 12*3600, 18*3600, 24*3600), labels = c("12am", "6am", "12pm"
  scale_y_continuous(labels=scales::comma_format(big.mark=',', decimal.mark = '.')) +
  labs(title="Number of Rides During the Day", subtitle="Split between Casual Riders and Members, and b
       x="Time Of Day", y="Number of Rides", fill="")
```



Duration of Rides

```
durationdata_df <- tripdata_df %>%
  mutate(ride_duration_minutes = as.double(difftime(ended_at, started_at, units="mins")))
```

```
cat("Ride duration in minutes:\n")
```

```
## Ride duration in minutes:
```

```
cat(sprintf("Min    = %6.1f minutes\n", min(durationdata_df$ride_duration_minutes)))
```

```
## Min    = -137.4 minutes
```

```
cat(sprintf("Max    = %6.1f minutes\n", max(durationdata_df$ride_duration_minutes)))
```

```
## Max    = 41629.2 minutes
```

It is seen that some ride durations are negative which of course shows that some of the duration data is wrong.

```
cat("Number of Duration < 0:", durationdata_df %>% filter(ride_duration_minutes < 0) %>% nrow(), "\n")
```

```
## Number of Duration < 0: 149
```

```
cat("Number of Duration = 0:", durationdata_df %>% filter(ride_duration_minutes == 0) %>% nrow(), "\n")
```

```
## Number of Duration = 0: 487
```

Because this is such a small number compared to the total number of rows, I remove all rows with duration less or equal to zero.

```r
durationdata_df <- durationdata_df %>% filter(0 < ride_duration_minutes)

cat("Number of Rows After Removing Rows with Non-Positive Durations:", nrow(durationdata_df))
```

```
## Number of Rows After Removing Rows with Non-Positive Durations: 5895237
```

```r
cat("Ride duration in minutes:\n")
```

```
## Ride duration in minutes:
```

```r
cat(sprintf("Min    = %6.1f minutes\n", min(durationdata_df$ride_duration_minutes)))
```

```
## Min    =    0.0 minutes
```

```r
cat(sprintf("Max    = %6.1f minutes\n", max(durationdata_df$ride_duration_minutes)))
```

```
## Max    = 41629.2 minutes
```

```r
cat(sprintf("Mean   = %6.1f minutes\n", mean(durationdata_df$ride_duration_minutes)))
```

```
## Mean   =   17.8 minutes
```

```r
cat(sprintf("Median = %6.1f minutes\n", median(durationdata_df$ride_duration_minutes)))
```

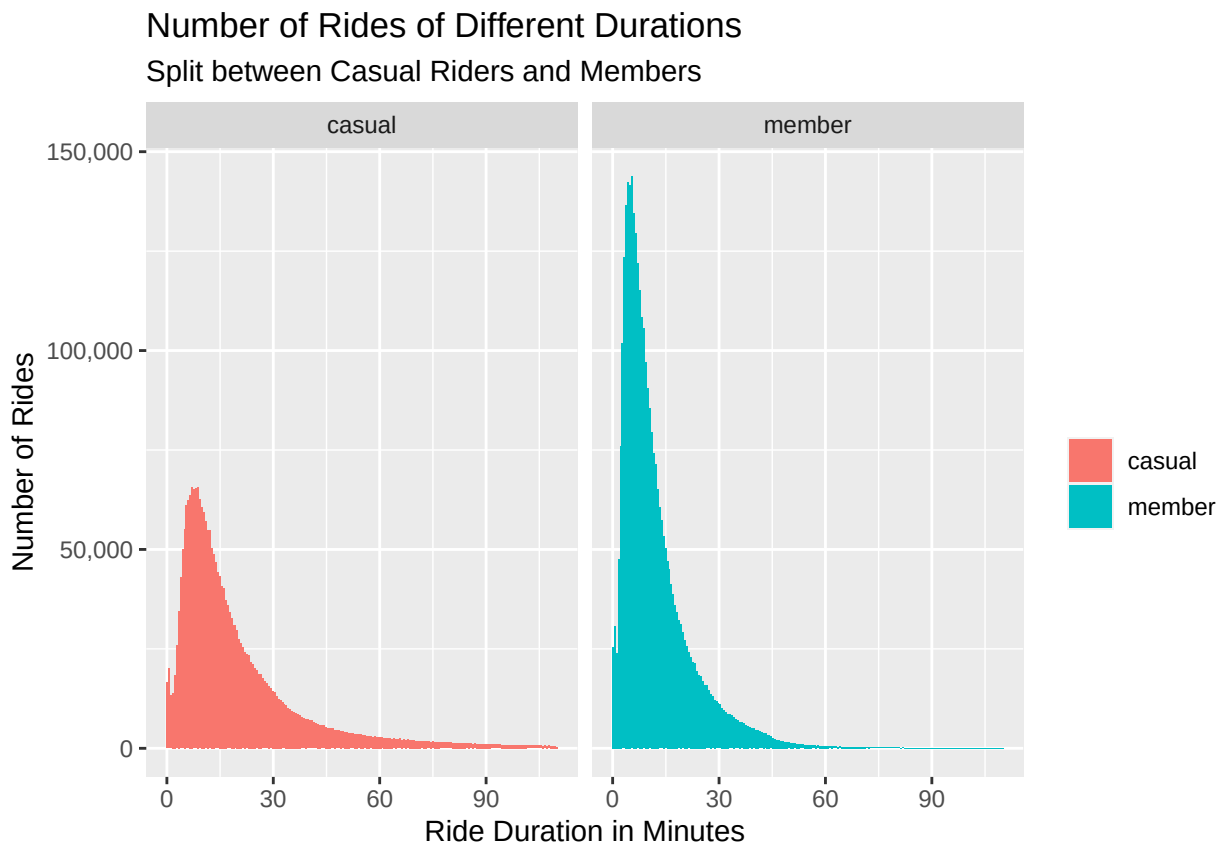
```
## Median =   10.9 minutes
```

```r
cat(sprintf("99%% percentile = %.1f minutes\n", quantile(durationdata_df$ride_duration_minutes, probs=0
```

```
## 99% percentile = 110.2 minutes
```

The max ride duration time is very much larger than the mean which again is larger than the median and this usually implies that the duration times are skewed with most short times and fewer larger and larger times. This is confirmed in the following chart where I cut-off the durations to 110 minutes because 99% of all durations are less than 10.8 minutes.

```r
durationdata_df %>%
  filter(as.double(ride_duration_minutes) < 110) %>%
  arrange(ride_duration_minutes) %>%
  ggplot(aes(x=ride_duration_minutes)) + geom_histogram(bins=200) +
  facet_grid(. ~ member_casual) + aes(fill=member_casual) +
  scale_y_continuous(labels=scales::comma_format(big.mark=',', decimal.mark = '.')) +
  labs(title="Number of Rides of Different Durations", subtitle="Split between Casual Riders and Member
       x="Ride Duration in Minutes", y="Number of Rides", fill="")
```



From the chart and from the following table of percentiles it is seen that members overall ride shorter rides than casual rides. The median ride duration is only about 9 minutes for members whereas it is about 14 minutes for casual riders.

```
durationdata_df %>%
  group_by(member_casual) %>%
  summarise(Min = min(ride_duration_minutes),
            Percentile20 = quantile(ride_duration_minutes, probs = .2),
            Percentile50 = quantile(ride_duration_minutes, probs = .5),
            Mean = mean(ride_duration_minutes),
            Percentile80 = quantile(ride_duration_minutes, probs = .8),
            Percentile90 = quantile(ride_duration_minutes, probs = .9),
            Percentile99 = quantile(ride_duration_minutes, probs = .99),
            Max = max(ride_duration_minutes))
```

```
## # A tibble: 2 x 9
##   member_casual    Min Percentile20 Perce~1  Mean Perce~2 Perce~3 Perce~4    Max
##   <fct>          <dbl>        <dbl>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>  <dbl>
## 1 casual        0.0167         7.12    14.4  24.7    30.6    48.3   149.  41629.
## 2 member        0.0167         4.57     9.02 12.7    18.0    25.5    52.5  1500.
## # ... with abbreviated variable names 1: Percentile50, 2: Percentile80,
## #   3: Percentile90, 4: Percentile99
```

Presentation

Slide 1: Business Task / Purpose

How Does Members and Casual Riders use Cyclistic Bikes Differently?

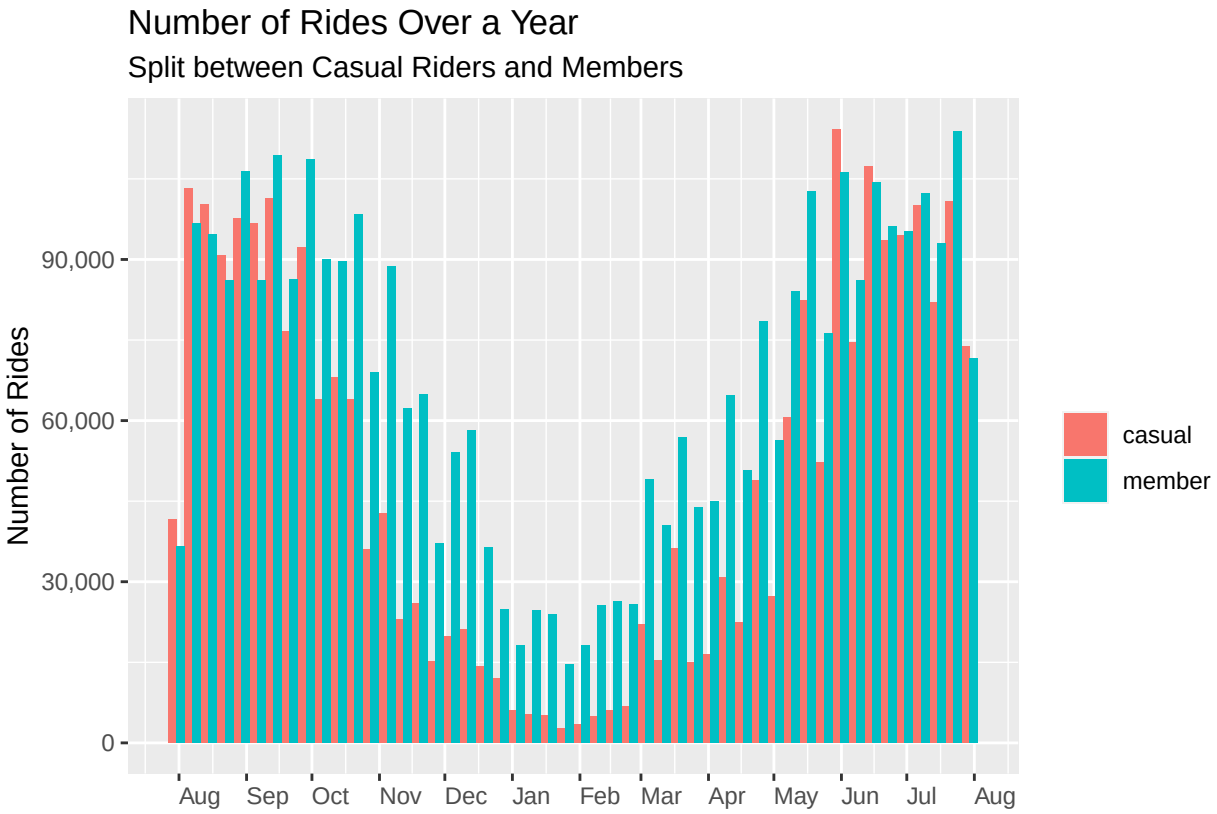
- author: Paw Hermansen
- date: Sept 11, 2022

Slide 2: Data

- The data is downloaded from https://divvy-tripdata.s3.amazonaws.com/index.html for the period August 2021 to July 2022, both included.
- The data is public and made available by Motivate International Inc. under a public license.
- All information about riders has been removed, i.e. no information about for example the frequency of rides for the two user types.
- The Data needed some clean-up but was otherwise good.

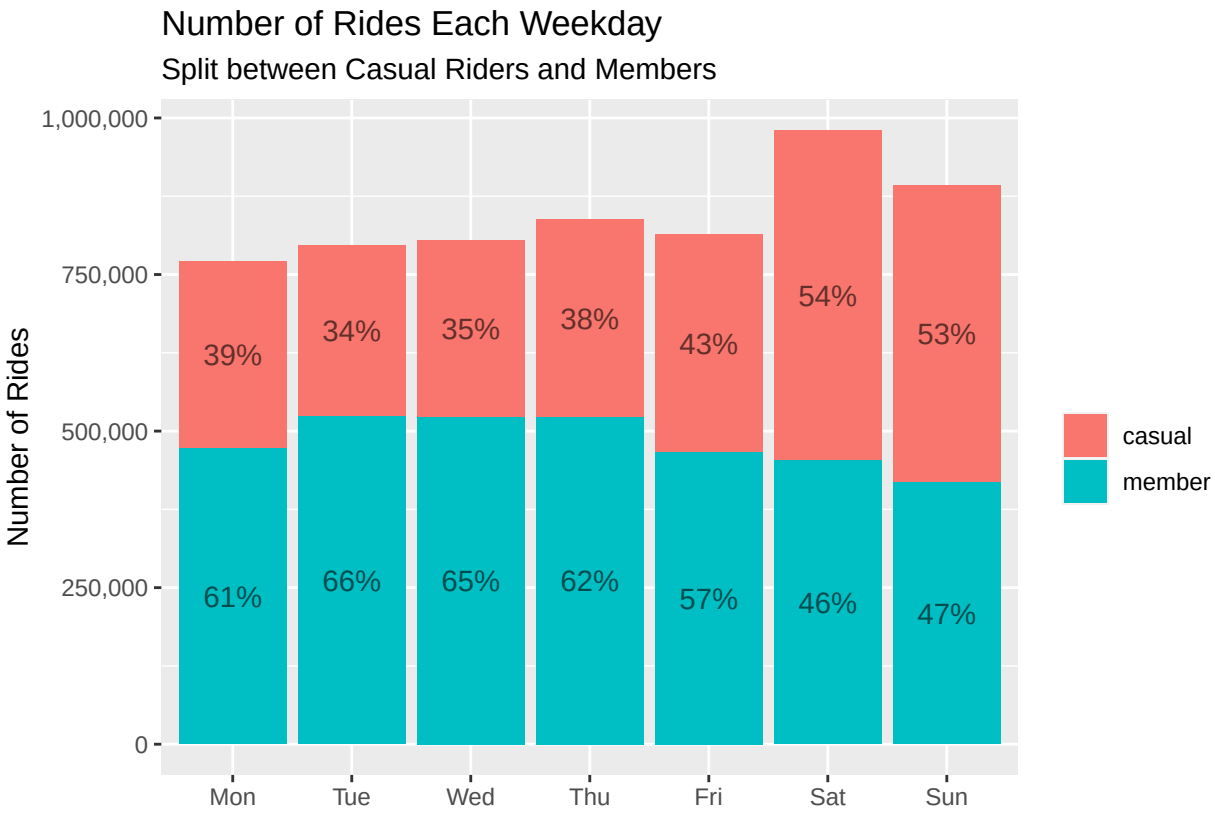
Slide 3: Seasonal Differences

- During summer months approximately half of the riders are casual rides and half are members.
- During winter months more than 80% of the rides are done by members.



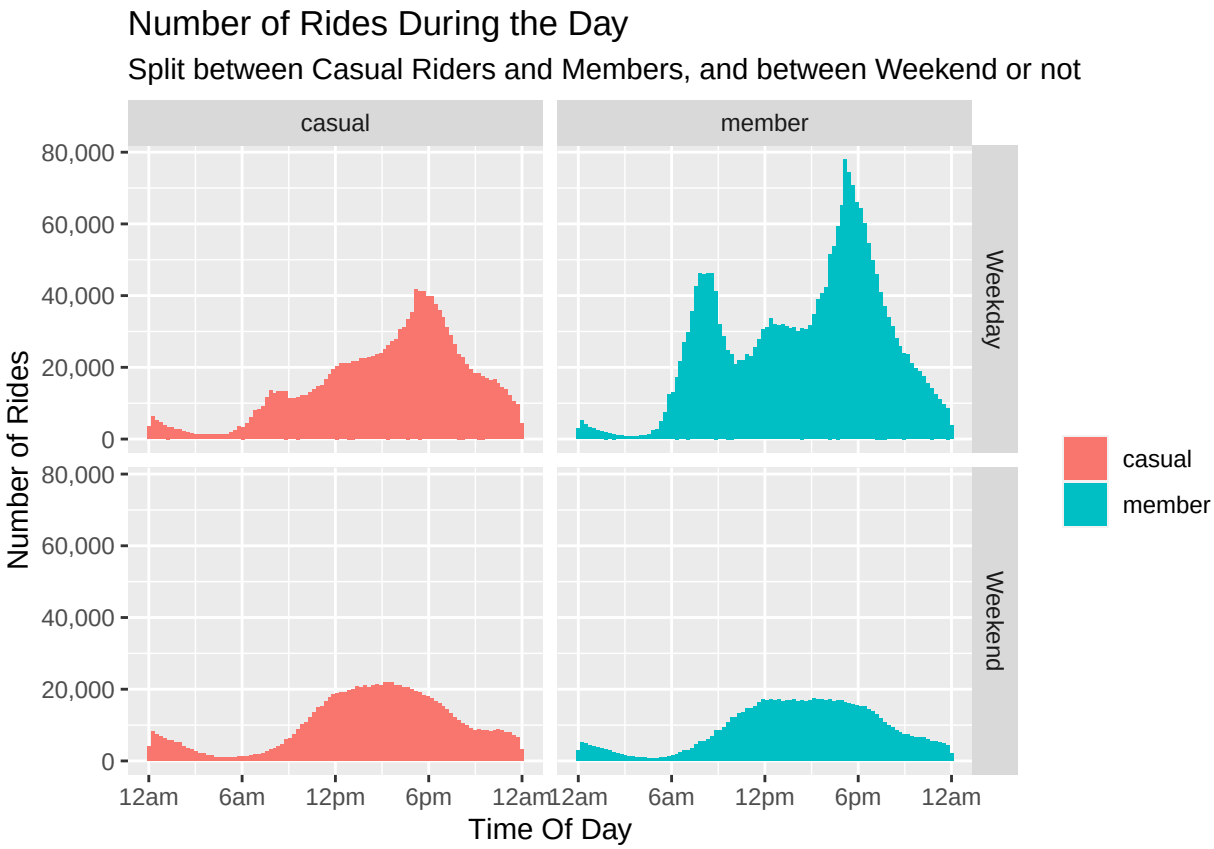
Slide 4: Weekly Differences

- During weekends more than half the rides are done by casual riders
- During weekdays more than half the rides are done by members



Slide 4: Daily Differences

- Members very clearly rides frequently in the hours just before and after work-hours on weekdays, Monday to Friday.
- Casual riders show this a lot less especially nearly no peak shows before work-hours.



Slide 5: Ride Durations

- For members the 20% shortest rides takes less than about 4 and a half minutes
- For casual riders the 20% shortest rides takes less than about 7 minutes

Duration in Minutes for Percentiles	20%	50%	80%
Casual Rider	7.1	14.9	30.6
Member	4.6	9.0	18.0

Slide 6: Summary

- Casual riders ride far less than members during winter months
- Casual riders ride more during weekends and members ride more during weekdays
- Casual riders rides are longer than member's rides.

Also worth mentioning:

- Casual riders do not ride to and from work as much as members appear to do. The difference is largest in the morning (before work-hours).