Cyclistic complete report

Nils

17-3-2022

Table of content

1.Purpose of this document

This document contains the Rstudio section of the steps undertaken to conduct my analysis of the Cyclistic capstone project of the Google Data Analytics professional course on Coursera. In short: the Google Data Analytics program provides a structured 5 months program for entry level data analyst positions.

I've divided the capstone project into 2 sections: Section 1: the Ask, Prepare, Process and Analyze sections are conducted below. The Share section (the visualization of the data) is done via Tableau. The end result of the project is uploaded to Github.
Note: If the reader of this document is only interested in the end product of this analysis please refer to the link above. This Rmarkdown document will show the complete record of my work with Rstudio (including mistakes I made and how I fixed those mistakes).

Scenario:

"I am working as a junior data analyst in the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. Therefore, my team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, the team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve my recommendations, so they must be backed up with compelling data insights and professional data visualizations."

Deliverables:

- 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.Top three recommendations

2.Ask

Identify the business task
The future success of the company depends on the conversion of casual riders to annual memberships. The purpose of this analysis is therefore to discover the key differences in the using of rental bikes by 2 different users of the bikes: annual members and casual riders.

Statement of the bussiness task:
How do annual members and casual riders use rental bikes differently?

Key stakeholders:
Cyclistic executive team, Director of Marketing (Lily Moreno), Marketing Analytics team.

3.Prepare

Gathering the datasets
The dataset consists of 12 months of inhouse data organised in 12 seperate CSV files. Made available via this link

Decription of the datasets:
Every csv file is organized in the same long data format consisting of the same 15 variables and over 20.000 observations.
Column names: ride ID, ride type, start/end time, ride length (in minutes), day of the week, starting point (code, name, and latitude/longitude), ending point (code, name, and latitude/longitude), and member/casual rider.

Preparing the datasets After downloading all the files to my laptop I've opened them up in excel and adjusted the column names so they were exactly the same, (same spelling, all lower case and no spatials etc.). Also added a ride_length column to the excel files as this seems useful in further analysis. Note: The datasets are too large to perform any meaningful work in Excel.

Verifying the credibility of the data:
* Reliable: the data is structured and organized in the same order
* Original: provided for by the Cyclistic company using it's own inhouse data
* Comprehensive: Full year worth of data detailing over 3 millions obeservations of 15 variables
* Current: april 2019 to march 2020)
* Cited: provided for by the Cyclistic company using it's own inhouse data

To proceed with the next steps of this analysis the excel files are uploaded to R:

setting up my environment

```
#setting up the working directory
setwd("C:/Users/nils_/Documents/Bike share case study/Processing_data/Version_1_ride_length-week_day")
```

```
#libraries I've used
library(tidyverse)
library(readxl)
```

```
library(here)
library(janitor)
library(dplyr)
library(lubridate)

#importing the datasets
y2020_04 <- read_excel("2020-04-divvy-tripdata.xlsx")
y2020_05 <- read_excel("2020-05-divvy-tripdata.xlsx")
y2020_06 <- read_excel("2020-06-divvy-tripdata.xlsx")
y2020_07 <- read_excel("2020-07-divvy-tripdata.xlsx")
y2020_08 <- read_excel("2020-08-divvy-tripdata.xlsx")
y2020_09 <- read_excel("2020-09-divvy-tripdata.xlsx")
y2020_10 <- read_excel("2020-10-divvy-tripdata.xlsx")
y2020_11 <- read_excel("2020-11-divvy-tripdata.xlsx")
y2020_12 <- read_excel("2020-12-divvy-tripdata.xlsx")
y2021_01 <- read_excel("2021-01-divvy-tripdata.xlsx")
y2021_02 <- read_excel("2021-02-divvy-tripdata.xlsx")
y2021_03 <- read_excel("2021-03-divvy-tripdata.xlsx")
```

4.Process

Combining the uploaded csv files into 1 dataframe using rbind:

```
full_year <- rbind(y2020_04, y2020_05, y2020_06, y2020_07, y2020_08, y2020_09, y2020_10, y2020_11, y202
```

Summary overview of the full_year dataframe:

```
colnames(full_year)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "week_day"           "ended_at"           "ride_length"
##  [7] "start_station_name" "start_station_id"   "end_station_name"
## [10] "end_station_id"     "start_lat"          "start_lng"
## [13] "end_lat"            "end_lng"            "member_casual"
```

```
skimr::skim_without_charts(full_year)
```

Table 1: Data summary

Name	full_year
Number of rows	3489748
Number of columns	15
Column type frequency:	
character	7
numeric	5
POSIXct	3

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1.00	6	23	0	3489539	0
rideable_type	0	1.00	11	13	0	3	0
start_station_name	122175	0.96	10	53	0	708	0
start_station_id	122801	0.96	1	35	0	1259	0
end_station_name	143242	0.96	10	53	0	706	0
end_station_id	143703	0.96	1	35	0	1259	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
week_day	0	1	4.207115e+08	2.208989e+09	-8.327526e+07	2	4	6	7.00000e+00
start_lat	0	1	1.931339e+15	5.220229e+15	1.170000e+02	41838556	41894448	41939365	4.20649e+16
start_lng	0	1	-8.786901e+14	2.629585e+15	-8.777460e+07	-87664358	-87635185	-87566445	-8.76000e+02
end_lat	4738	1	2.143787e+15	5.460677e+15	1.170000e+02	41857611	41906866	41926036	4.20649e+16
end_lng	4738	1	-1.128163e+15	2.920950e+15	-8.777470e+07	-87678481	-87641066	-87599861	-8.76000e+02

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2020-04-01 00:00:00	2021-03-31 23:59:00	2020-08-29 14:50:00	395258
ended_at	0	1	2020-04-01 00:10:00	2021-04-06 11:00:00	2020-08-29 15:21:00	396388
ride_length	379	1	1899-12-30 22:00:00	1900-02-09 18:40:00	1899-12-31 00:15:00	2887

```
glimpse(full_year)
```

```
## Rows: 3,489,748
## Columns: 15
## $ ride_id            <chr> "A847FADBBC638E45", "5405B80E996FF60D", "5DD24A79A4~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <dttm> 2020-04-26 17:45:00, 2020-04-17 17:08:00, 2020-04-~
## $ week_day           <dbl> 7, 5, 3, 2, 6, 4, 4, 2, 3, 6, 6, 6, 5, 6, 1, 6, 7, ~
## $ ended_at           <dttm> 2020-04-26 18:12:00, 2020-04-17 17:17:00, 2020-04-~
## $ ride_length        <dttm> 1899-12-31 00:27:00, 1899-12-31 00:09:00, 1899-12-~
## $ start_station_name <chr> "Eckhart Park", "Drake Ave & Fullerton Ave", "McClu~
```

```
## $ start_station_id   <chr> "86", "503", "142", "216", "125", "173", "35", "434~
## $ end_station_name   <chr> "Lincoln Ave & Diversey Pkwy", "Kosciuszko Park", "~
## $ end_station_id     <chr> "152", "499", "255", "657", "323", "35", "635", "38~
## $ start_lat          <dbl> 418964, 419244, 418945, 41903, 418902, 418969, 4189~
## $ start_lng          <dbl> -87661, -877154, -876179, -876975, -876262, -876217~
## $ end_lat            <dbl> 419322, 419306, 418679, 418992, 419695, 418923, 418~
## $ end_lng            <dbl> -876586, -877238, -87623, -876722, -876547, -87612,~
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "~
```

```
head(full_year)
```

```
## # A tibble: 6 x 15
##   ride_id         rideable_type started_at          week_day ended_at
##   <chr>           <chr>         <dttm>                 <dbl> <dttm>
## 1 A847FADBBC638E~ docked_bike   2020-04-26 17:45:00        7 2020-04-26 18:12:00
## 2 5405B80E996FF6~ docked_bike   2020-04-17 17:08:00        5 2020-04-17 17:17:00
## 3 5DD24A79A4E006~ docked_bike   2020-04-01 17:54:00        3 2020-04-01 18:08:00
## 4 2A59BBDF5CDBA7~ docked_bike   2020-04-07 12:50:00        2 2020-04-07 13:02:00
## 5 27AD306C119C61~ docked_bike   2020-04-18 10:22:00        6 2020-04-18 11:15:00
## 6 356216E875132F~ docked_bike   2020-04-30 17:55:00        4 2020-04-30 18:01:00
## # ... with 10 more variables: ride_length <dttm>, 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>
```

Looking up how many missing values there are in this dataset (full_year)

```
sum(is.na(full_year))
```

```
## [1] 541776
```

Note: 541,776 missing values out of a total of 3,489,748 this amounts to 15.5% of the total dataset

to further specify in which columns the missing values are concentrated:

```
colSums(is.na(full_year))
```

```
##            ride_id      rideable_type         started_at           week_day
##                  0                  0                  0                  0
##          ended_at        ride_length start_station_name   start_station_id
##                  0                379             122175             122801
##   end_station_name     end_station_id          start_lat          start_lng
##             143242             143703                  0                  0
##            end_lat            end_lng      member_casual
##               4738               4738                  0
```

Note: All missing values reside in the ride_length, station names/id and end_lat/lng columns.

Dropping the missing values from the dataset

```
full_year_cleaned <- na.omit(full_year)
sum(is.na(full_year_cleaned))
```

Note: after dropping the missing values 3294375 out of 3489748 remain. meaning that 195373 missing values are removed (5.6%)

New summaries

Table 5: Data summary

Name	full_year_cleaned
Number of rows	3294375
Number of columns	15
Column type frequency:	
character	7
numeric	5
POSIXct	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1	6	23	0	3294375	0
rideable_type	0	1	11	13	0	3	0
start_station_name	0	1	10	53	0	702	0
start_station_id	0	1	1	35	0	1256	0
end_station_name	0	1	10	53	0	704	0
end_station_id	0	1	1	35	0	1258	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
week_day	0	1	-7.616110e+07	4.030125e+08	-2.208989e+09	2	4	6	7.00000e+00
start_lat	0	1	1.779823e+17	5.905215e+15	4.188000e+01	41860384	41894722	41937582	4.20649e+16
start_lng	0	1	-8.119279e+14	2.538026e+15	-8.777460e+07	-87663913	-87635689	-87607267	-8.76300e+03
end_lat	0	1	2.061088e+15	5.285786e+15	4.190000e+01	41867888	41909668	41949475	4.20649e+16
end_lng	0	1	-1.102083e+15	2.890793e+15	-8.777470e+07	-87623776	-87643909	-87613533	-8.75400e+03

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2020-04-01 00:00:00	2021-03-31 23:59:00	2020-08-24 22:17:00	386320
ended_at	0	1	2020-04-01 00:10:00	2021-04-06 11:00:00	2020-08-24 22:57:00	387294
ride_length	0	1	1899-12-30 22:00:00	1900-02-09 18:40:00	1899-12-31 00:15:00	2877

```
## Rows: 3,294,375
## Columns: 15
## $ ride_id            <chr> "A847FADBBC638E45", "5405B80E996FF60D", "5DD24A79A4~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <dttm> 2020-04-26 17:45:00, 2020-04-17 17:08:00, 2020-04-~
## $ week_day           <dbl> 7, 5, 3, 2, 6, 4, 4, 2, 3, 6, 6, 6, 5, 6, 1, 6, 7, ~
## $ ended_at           <dttm> 2020-04-26 18:12:00, 2020-04-17 17:17:00, 2020-04-~
## $ ride_length        <dttm> 1899-12-31 00:27:00, 1899-12-31 00:09:00, 1899-12-~
## $ start_station_name <chr> "Eckhart Park", "Drake Ave & Fullerton Ave", "McClu~
## $ start_station_id   <chr> "86", "503", "142", "216", "125", "173", "35", "434~
## $ end_station_name   <chr> "Lincoln Ave & Diversey Pkwy", "Kosciuszko Park", "~
## $ end_station_id     <chr> "152", "499", "255", "657", "323", "35", "635", "38~
## $ start_lat          <dbl> 418964, 419244, 418945, 41903, 418902, 418969, 4189~
## $ start_lng          <dbl> -87661, -877154, -876179, -876975, -876262, -876217~
## $ end_lat            <dbl> 419322, 419306, 418679, 418992, 419695, 418923, 418~
## $ end_lng            <dbl> -876586, -877238, -87623, -876722, -876547, -87612,~
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "~
```

next steps: finding other irregularities within the dataset

the station_id columns (start and end both) are in the wrong data type (chr) and need to be converted to numeric to perform analysis

```
class(full_year_cleaned$start_station_id)
```

```
## [1] "character"
```

```
class(full_year_cleaned$end_station_id)
```

```
## [1] "character"
```

Note: Some observations include Letters, this is probably why R converted the column into a Character string. These values are useless because there is no way to interpret them. These values need to be omitted from the dataframe

Step 1 converting these columns into numeric:

```
full_year_cleaned_V01 <- mutate(full_year_cleaned, start_station_id = as.numeric(start_station_id),
                   end_station_id = as.numeric(end_station_id))
```

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

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

Step 2 checking the result:

```
class(full_year_cleaned_V01$start_station_id)
```

```
## [1] "numeric"
```

```
class(full_year_cleaned_V01$end_station_id)
```

```
## [1] "numeric"
```

Step 3 Checking for added n.a values (there should be some now because of the conversion)

```
sum(is.na(full_year_cleaned_V01))
```

Step 4 Removing the n.a values from the dataset and assigning a new df to keep track of the changes

```
full_year_cleaned_V02 <- na.omit(full_year_cleaned_V01)
```

Step 5 Checking the result:

```
sum(is.na(full_year_cleaned_V02))
```

```
## [1] 0
```

```
colSums(is.na(full_year_cleaned_V02))
```

```
##            ride_id      rideable_type         started_at           week_day
##                  0                  0                  0                  0
##          ended_at        ride_length start_station_name   start_station_id
##                  0                  0                  0                  0
##   end_station_name     end_station_id          start_lat          start_lng
##                  0                  0                  0                  0
##            end_lat            end_lng      member_casual
##                  0                  0                  0
```

Other problems that need to be addressed:

The dataset can only be aggregated at the ride-level. This is too granular. Adding some additional columns of data such as day, month, year would improve the analysis and provide additional opportunities to aggregate the data

```
full_year_cleaned_V02$month <- format(as.Date(full_year_cleaned_V02$date), "%m")
full_year_cleaned_V02$month_day <- format(as.Date(full_year_cleaned_V02$date), "%d")
full_year_cleaned_V02$year <- format(as.Date(full_year_cleaned_V02$date), "%Y")
full_year_cleaned_V02$day_of_week <- format(as.Date(full_year_cleaned_V02$date), "%A")
glimpse(full_year_cleaned_V02)
```

Note: Because of the settings of my laptop R automatically formats the days of the weeks in Dutch. Have not found a workaround to change this!

Fiddling around I created 2 extra variables (columns): weekdays and day, these can now be dropped

```
full_year_cleaned_V02$day = NULL
full_year_cleaned_V02$weekdays = NULL
```

From the original data set I created an extra column in excel called week_day, this column can also be removed as it replaced by the data added above

```
full_year_cleaned_V02$week_day = NULL
```

Checking the result

```
glimpse(full_year_cleaned_V02)
```

```
## Rows: 2,944,638
## Columns: 19
## $ ride_id            <chr> "A847FADBBC638E45", "5405B80E996FF60D", "5DD24A79A4~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <dttm> 2020-04-26 17:45:00, 2020-04-17 17:08:00, 2020-04-~
## $ ended_at           <dttm> 2020-04-26 18:12:00, 2020-04-17 17:17:00, 2020-04-~
## $ ride_length        <dttm> 1899-12-31 00:27:00, 1899-12-31 00:09:00, 1899-12-~
## $ start_station_name <chr> "Eckhart Park", "Drake Ave & Fullerton Ave", "McClu~
## $ start_station_id   <dbl> 86, 503, 142, 216, 125, 173, 35, 434, 627, 377, 508~
## $ end_station_name   <chr> "Lincoln Ave & Diversey Pkwy", "Kosciuszko Park", "~
## $ end_station_id     <dbl> 152, 499, 255, 657, 323, 35, 635, 382, 359, 508, 37~
## $ start_lat          <dbl> 418964, 419244, 418945, 41903, 418902, 418969, 4189~
## $ start_lng          <dbl> -87661, -877154, -876179, -876975, -876262, -876217~
## $ end_lat            <dbl> 419322, 419306, 418679, 418992, 419695, 418923, 418~
## $ end_lng            <dbl> -876586, -877238, -87623, -876722, -876547, -87612,~
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "~
## $ date               <date> 2020-04-26, 2020-04-17, 2020-04-01, 2020-04-07, 20~
## $ month              <chr> "04", "04", "04", "04", "04", "04", "04", "04", "04~
## $ month_day          <chr> "26", "17", "01", "07", "18", "30", "02", "07", "15~
## $ year               <chr> "2020", "2020", "2020", "2020", "2020", "2020", "20~
## $ day_of_week        <chr> "zondag", "vrijdag", "woensdag", "dinsdag", "zaterd~
```

saving these results into a new dataframe V03 before proceeding into the next step

```
full_year_cleaned_V03 <- full_year_cleaned_V02
```

Problems continued:

Before uploading the excel files to Rstudio I've created an extra column to calculate ride_length. This ride length_column holds a date PLUS the time, I only want to preserve the time part of the column. Alternative: used the timediff function to calculate the ride_length with Rstudio.
Step 1: dropping the existing ride_length column

```
full_year_cleaned_V03$ride_length = NULL
```

Step 2: adding the new ride_length column

```
full_year_cleaned_V03$ride_length <- difftime(full_year_cleaned_V03$ended_at,full_year_cleaned_V03$star
```

The new structure of the columns

```
str(full_year_cleaned_V03)
```

```
## tibble [2,944,638 x 19] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:2944638] "A847FADBBC638E45" "5405B80E996FF60D" "5DD24A79A4E006F4" "2A5
##  $ rideable_type     : chr [1:2944638] "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : POSIXct[1:2944638], format: "2020-04-26 17:45:00" "2020-04-17 17:08:00" ...
##  $ ended_at          : POSIXct[1:2944638], format: "2020-04-26 18:12:00" "2020-04-17 17:17:00" ...
##  $ start_station_name: chr [1:2944638] "Eckhart Park" "Drake Ave & Fullerton Ave" "McClurg Ct & Erie
##  $ start_station_id  : num [1:2944638] 86 503 142 216 125 173 35 434 627 377 ...
##  $ end_station_name  : chr [1:2944638] "Lincoln Ave & Diversey Pkwy" "Kosciuszko Park" "Indiana Ave &
##  $ end_station_id    : num [1:2944638] 152 499 255 657 323 35 635 382 359 508 ...
##  $ start_lat         : num [1:2944638] 418964 419244 418945 41903 418902 ...
##  $ start_lng         : num [1:2944638] -87661 -877154 -876179 -876975 -876262 ...
##  $ end_lat           : num [1:2944638] 419322 419306 418679 418992 419695 ...
##  $ end_lng           : num [1:2944638] -876586 -877238 -87623 -876722 -876547 ...
##  $ member_casual     : chr [1:2944638] "member" "member" "member" "member" ...
##  $ date              : Date[1:2944638], format: "2020-04-26" "2020-04-17" ...
##  $ month             : chr [1:2944638] "04" "04" "04" "04" ...
##  $ month_day         : chr [1:2944638] "26" "17" "01" "07" ...
##  $ year              : chr [1:2944638] "2020" "2020" "2020" "2020" ...
##  $ day_of_week       : chr [1:2944638] "zondag" "vrijdag" "woensdag" "dinsdag" ...
##  $ ride_length       : 'difftime' num [1:2944638] 1620 540 840 720 ...
##   ..- attr(*, "units")= chr "secs"
##  - attr(*, "na.action")= 'omit' Named int [1:349737] 2848584 2848585 2848586 2848587 2848588 2848591
##   ..- attr(*, "names")= chr [1:349737] "2848584" "2848585" "2848586" "2848587" ...
```

Step 3: converting "ride_length" from Factor to numeric in order to run calculations on the data

```
is.factor(full_year_cleaned_V03$ride_length)
full_year_cleaned_V03$ride_length <- as.numeric(as.character(full_year_cleaned_V03$ride_length))
is.numeric(full_year_cleaned_V03$ride_length)
```

Last remaining problems:

The dataframe includes entries when bikes were taken out of docks and checked for quality or ride_length was negative

```
full_year_cleaned_V04 <- full_year_cleaned_V03[!(full_year_cleaned_V03$start_station_name == "HQ QR" |
```

Final steps

```
sum(is.na(full_year_cleaned_V04))
str(full_year_cleaned_V04)
```

5.Analysis

Conducting descriptive analysis

Descriptive analysis on ride_length (in minutes)

```
mean(full_year_cleaned_V04$ride_length/60) #straight average (total ride length / rides)
```

```
## [1] 29.50396
```

```
median(full_year_cleaned_V04$ride_length/60) #midpoint number in the ascending array of ride lengths
```

```
## [1] 16
```

```
max(full_year_cleaned_V04$ride_length/60) #longest ride
```

```
## [1] 58720
```

```
min(full_year_cleaned_V04$ride_length/60) #shortest ride
```

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

Note: Discovered that the max ride_length is longer than 24 hours! (58720 minutes: >40 days!) This skews the results of the analysis.

Setting a limit to the max amount of time that a bike could be used to 24 hours (<86400 seconds)

```
full_year_cleaned_V04 <- full_year_cleaned_V04[!(full_year_cleaned_V04$ride_length>86400),]
```

The adjusted descriptive analysis on ride_length (in minutes)

```
mean(full_year_cleaned_V04$ride_length/60) #straight average (total ride length / rides)
```

```
## [1] 25.42589
```

```
median(full_year_cleaned_V04$ride_length/60) #midpoint number in the ascending array of ride lengths
```

```
## [1] 16
```

```
max(full_year_cleaned_V04$ride_length/60) #longest ride
```

```
## [1] 1440
```

```
min(full_year_cleaned_V04$ride_length/60) #shortest ride
```

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

Comparing members and casual users

```
aggregate(full_year_cleaned_V04$ride_length/60 ~ full_year_cleaned_V04$member_casual, FUN = mean)
```

```
##   full_year_cleaned_V04$member_casual full_year_cleaned_V04$ride_length/60
## 1                             casual                              37.87731
## 2                             member                              16.08582
```

```
aggregate(full_year_cleaned_V04$ride_length/60 ~ full_year_cleaned_V04$member_casual, FUN = median)
```

```
##   full_year_cleaned_V04$member_casual full_year_cleaned_V04$ride_length/60
## 1                             casual                                   22
## 2                             member                                   12
```

```
aggregate(full_year_cleaned_V04$ride_length/60 ~ full_year_cleaned_V04$member_casual, FUN = max)
```

```
##   full_year_cleaned_V04$member_casual full_year_cleaned_V04$ride_length/60
## 1                             casual                                 1440
## 2                             member                                 1440
```

```
aggregate(full_year_cleaned_V04$ride_length/60 ~ full_year_cleaned_V04$member_casual, FUN = min)
```

```
##   full_year_cleaned_V04$member_casual full_year_cleaned_V04$ride_length/60
## 1                             casual                                    1
## 2                             member                                    1
```

The average ride time per day for members vs casual users

```
aggregate(full_year_cleaned_V04$ride_length/60 ~ full_year_cleaned_V04$member_casual + full_year_cleaned
```

```
##    full_year_cleaned_V04$member_casual full_year_cleaned_V04$day_of_week
## 1                              casual                           dinsdag
## 2                              member                           dinsdag
## 3                              casual                         donderdag
## 4                              member                         donderdag
## 5                              casual                           maandag
## 6                              member                           maandag
## 7                              casual                           vrijdag
## 8                              member                           vrijdag
## 9                              casual                          woensdag
## 10                             member                          woensdag
## 11                             casual                          zaterdag
## 12                             member                          zaterdag
## 13                             casual                            zondag
## 14                             member                            zondag
```

```
##    full_year_cleaned_V04$ride_length/60
## 1                              35.20001
## 2                              15.18351
## 3                              34.55196
## 4                              15.28757
## 5                              37.51911
## 6                              15.19402
## 7                              36.25668
## 8                              15.83345
## 9                              33.99380
## 10                             15.15491
## 11                             40.39012
## 12                             17.84335
## 13                             42.00070
## 14                             17.98866
```

Note: : The days of the week are out of order. to fix this:

```
full_year_cleaned_V04$day_of_week <- ordered(full_year_cleaned_V04$day_of_week, levels=c("maandag", "di
```

The correctly ordered day of the week average ride time per day for members vs casual users

```
aggregate(full_year_cleaned_V04$ride_length/60 ~ full_year_cleaned_V04$member_casual + full_year_cleane
```

```
##    full_year_cleaned_V04$member_casual full_year_cleaned_V04$day_of_week
## 1                              casual                           maandag
## 2                              member                           maandag
## 3                              casual                           dinsdag
## 4                              member                           dinsdag
## 5                              casual                          woensdag
## 6                              member                          woensdag
## 7                              casual                         donderdag
## 8                              member                         donderdag
## 9                              casual                           vrijdag
## 10                             member                           vrijdag
## 11                             casual                          zaterdag
## 12                             member                          zaterdag
## 13                             casual                            zondag
## 14                             member                            zondag
##    full_year_cleaned_V04$ride_length/60
## 1                              37.51911
## 2                              15.19402
## 3                              35.20001
## 4                              15.18351
## 5                              33.99380
## 6                              15.15491
## 7                              34.55196
## 8                              15.28757
## 9                              36.25668
## 10                             15.83345
## 11                             40.39012
## 12                             17.84335
## 13                             42.00070
## 14                             17.98866
```

Analysis of the ridership data by type and weekday

```
full_year_cleaned_V04 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(member_casual, weekday) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()                              #calculates the number of rides and average
            ,average_duration = mean(ride_length/60)) %>%       # calculates the average duration
  arrange(member_casual, weekday)                               # sorts
```

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

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        zo               230864             42.0
##  2 casual        ma               129072             37.5
##  3 casual        di               123903             35.2
##  4 casual        wo               136385             34.0
##  5 casual        do               146071             34.6
##  6 casual        vr               184818             36.3
##  7 casual        za               294862             40.4
##  8 member        zo               217907             18.0
##  9 member        ma               212884             15.2
## 10 member        di               225007             15.2
## 11 member        wo               244213             15.2
## 12 member        do               245153             15.3
## 13 member        vr               251448             15.8
## 14 member        za               264421             17.8
```

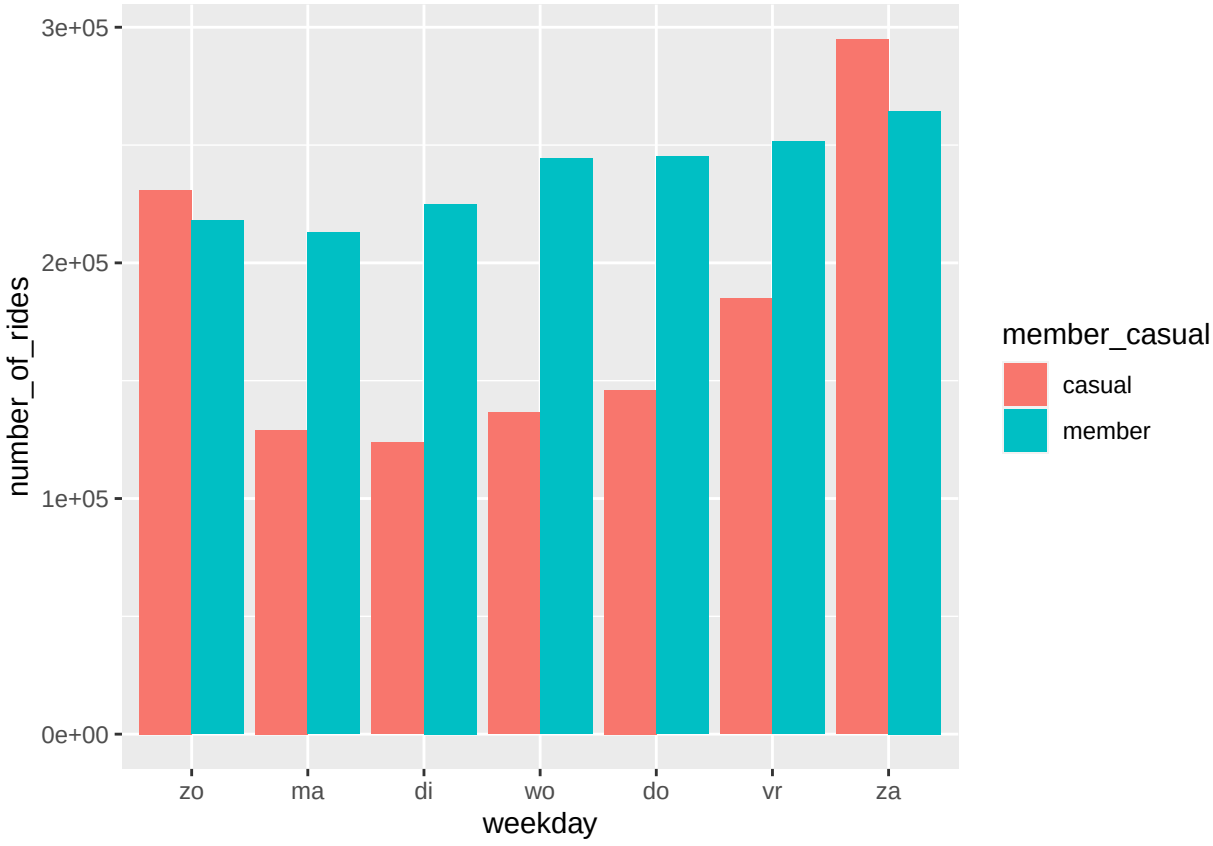
6.Share

Note: Below follow a few diagrams to check if this cleaned dataset yields results that I can further explore in Tableau

Number of rides members and casuals

```
full_year_cleaned_V04 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length/60)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

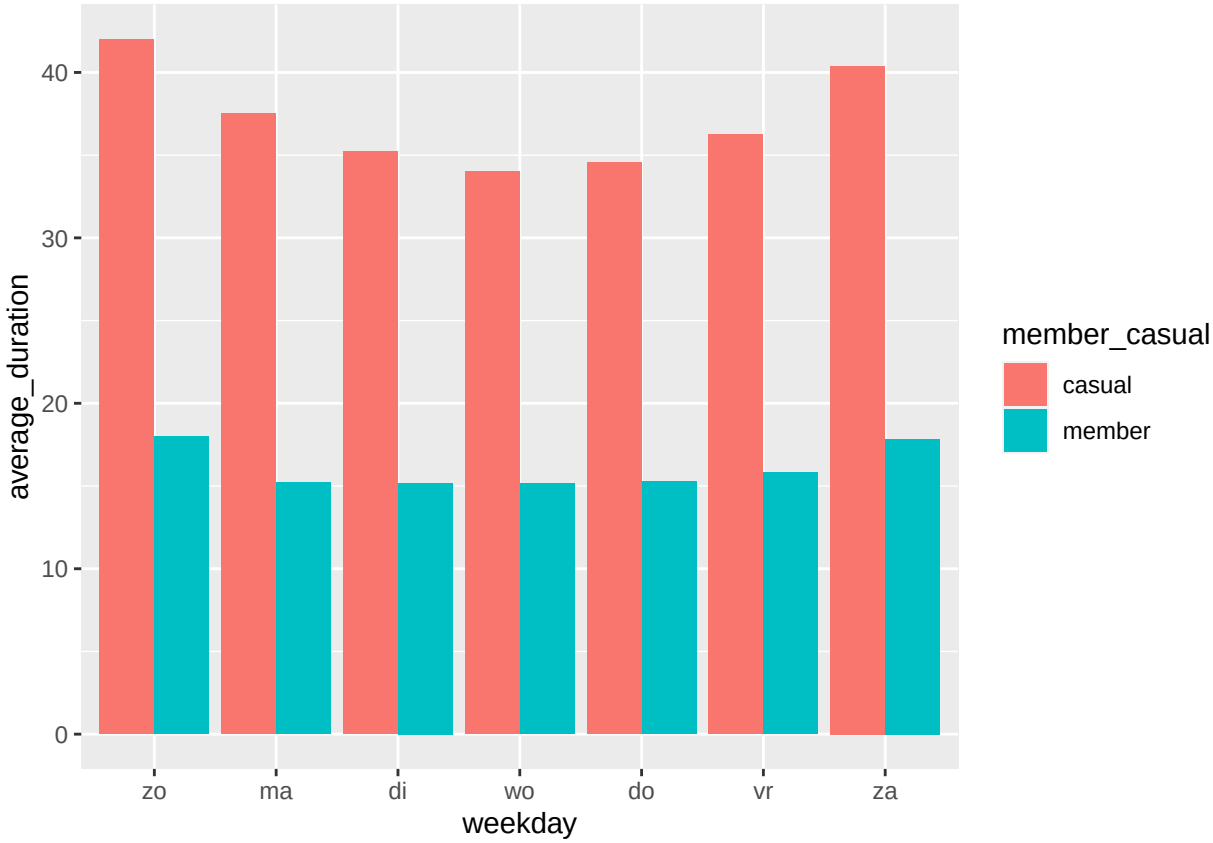
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```



Average duration of trips

```
full_year_cleaned_V04 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length/60)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

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



Exporting this dataframe to a CSV file:

Uploading the csv file to Tableau for further analysis:

Link to the slide show on tableau

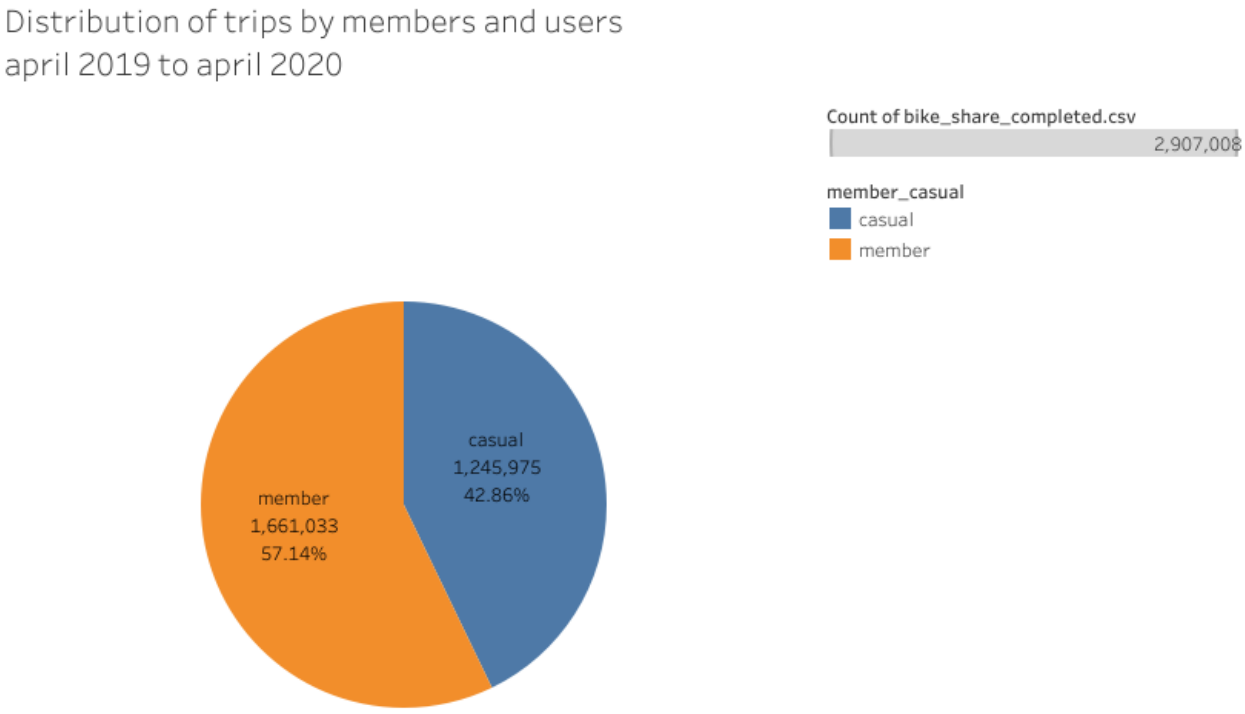


Figure 1: Alt text

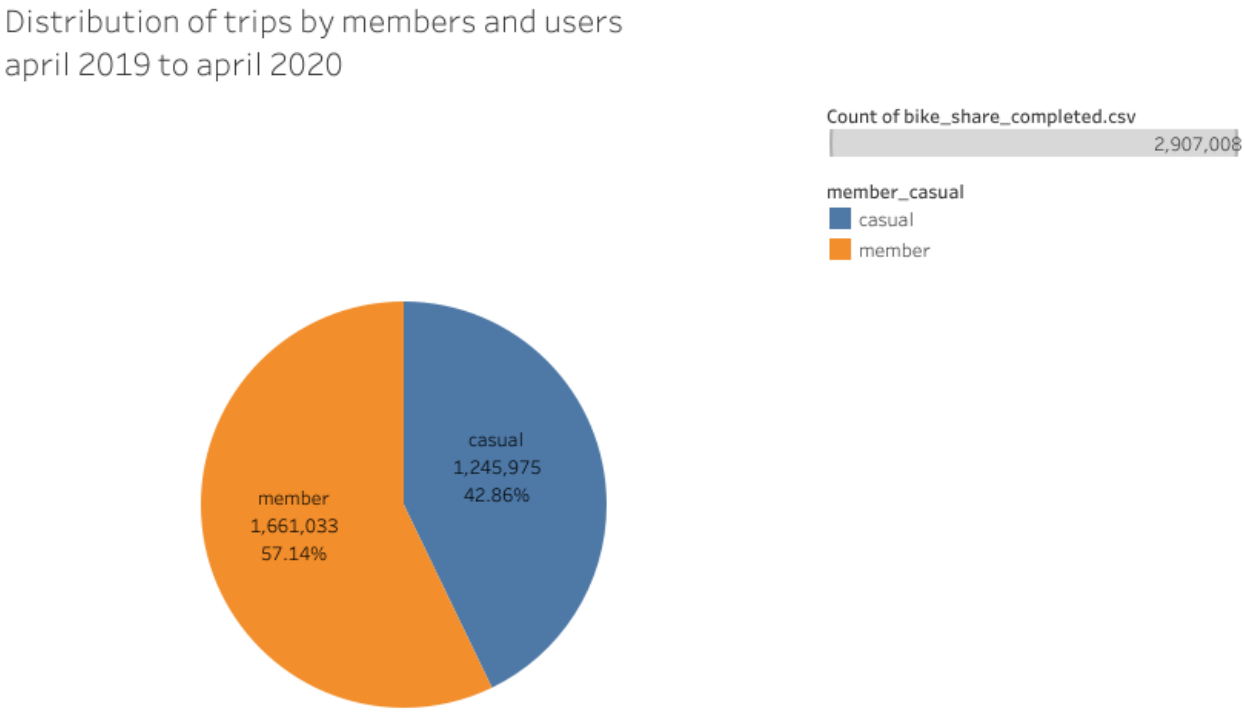
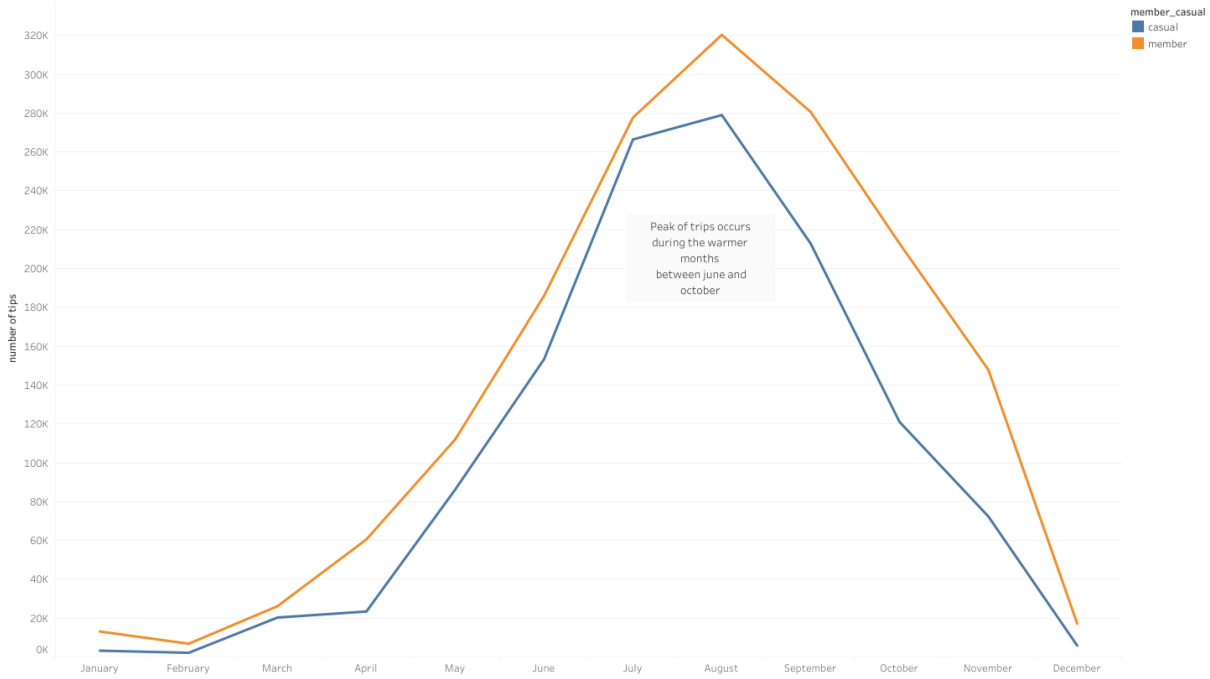
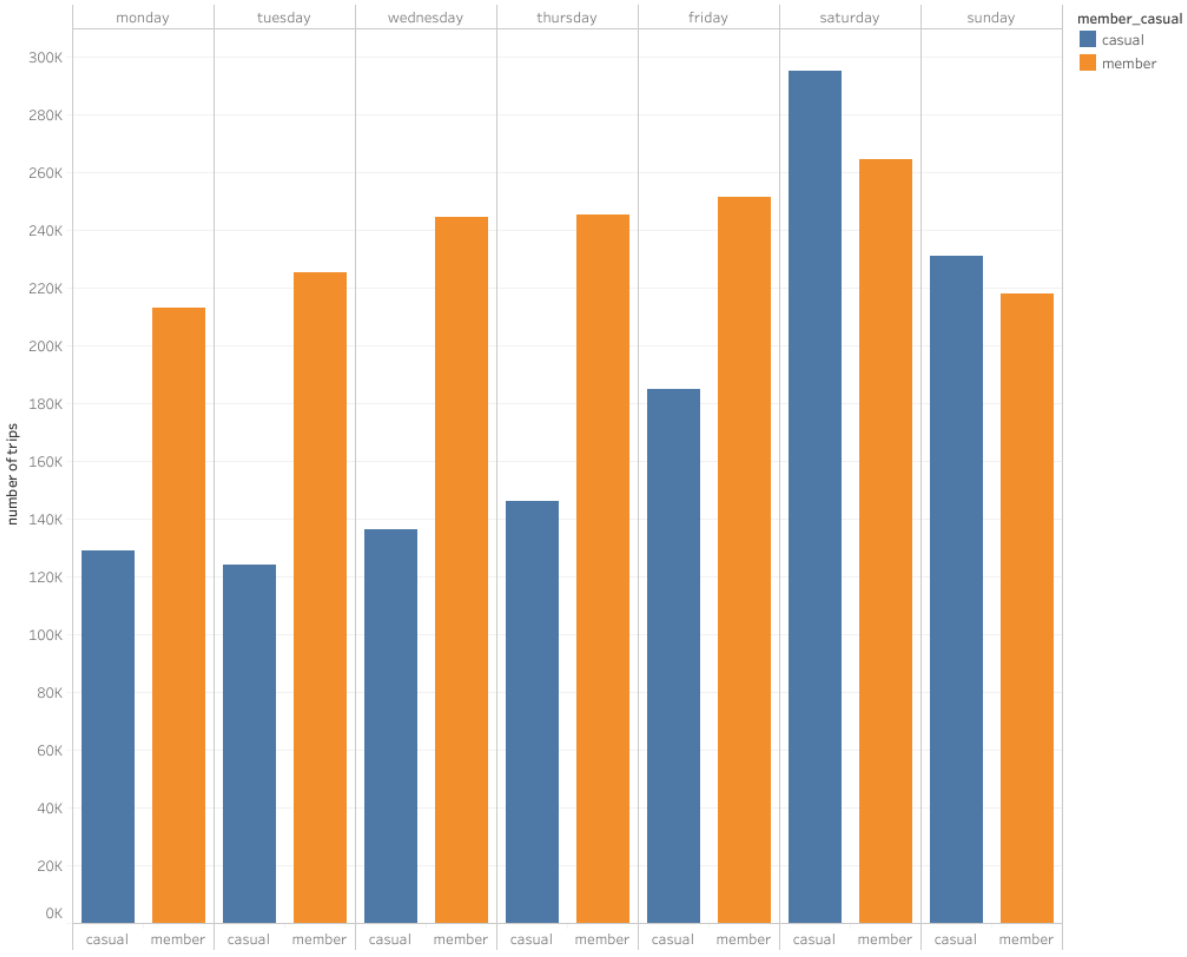


Figure 2: Alt text

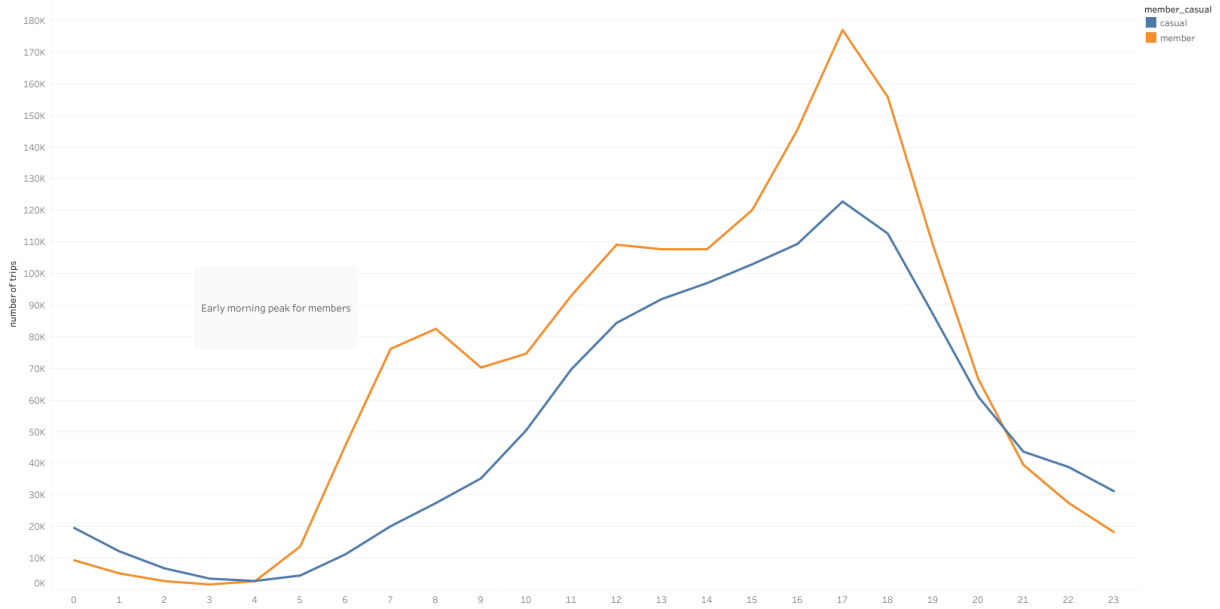
Number of trips per month



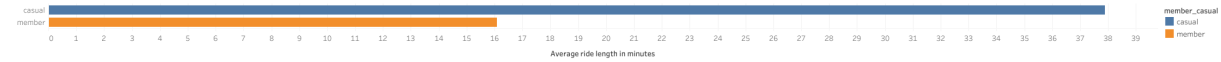
Amount of trips per day of the week



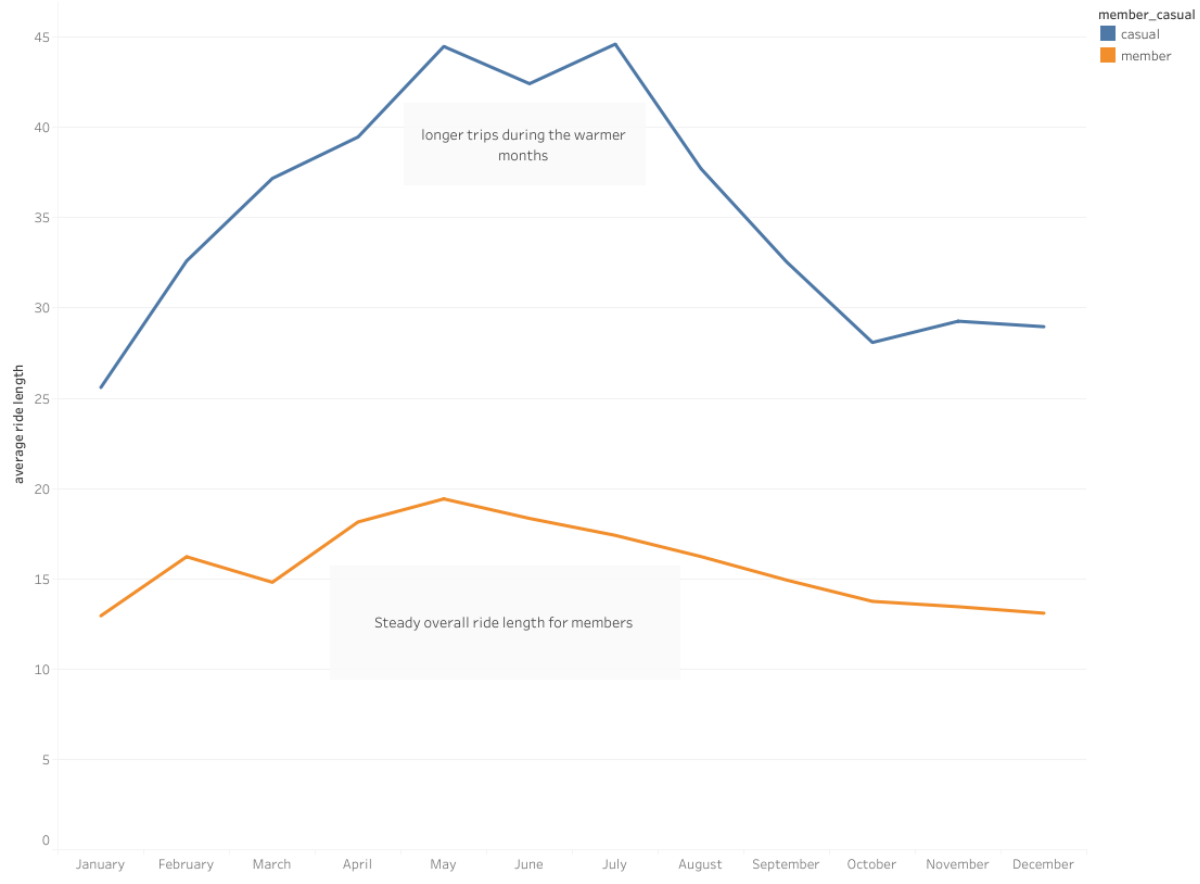
Starting times of trips per hour



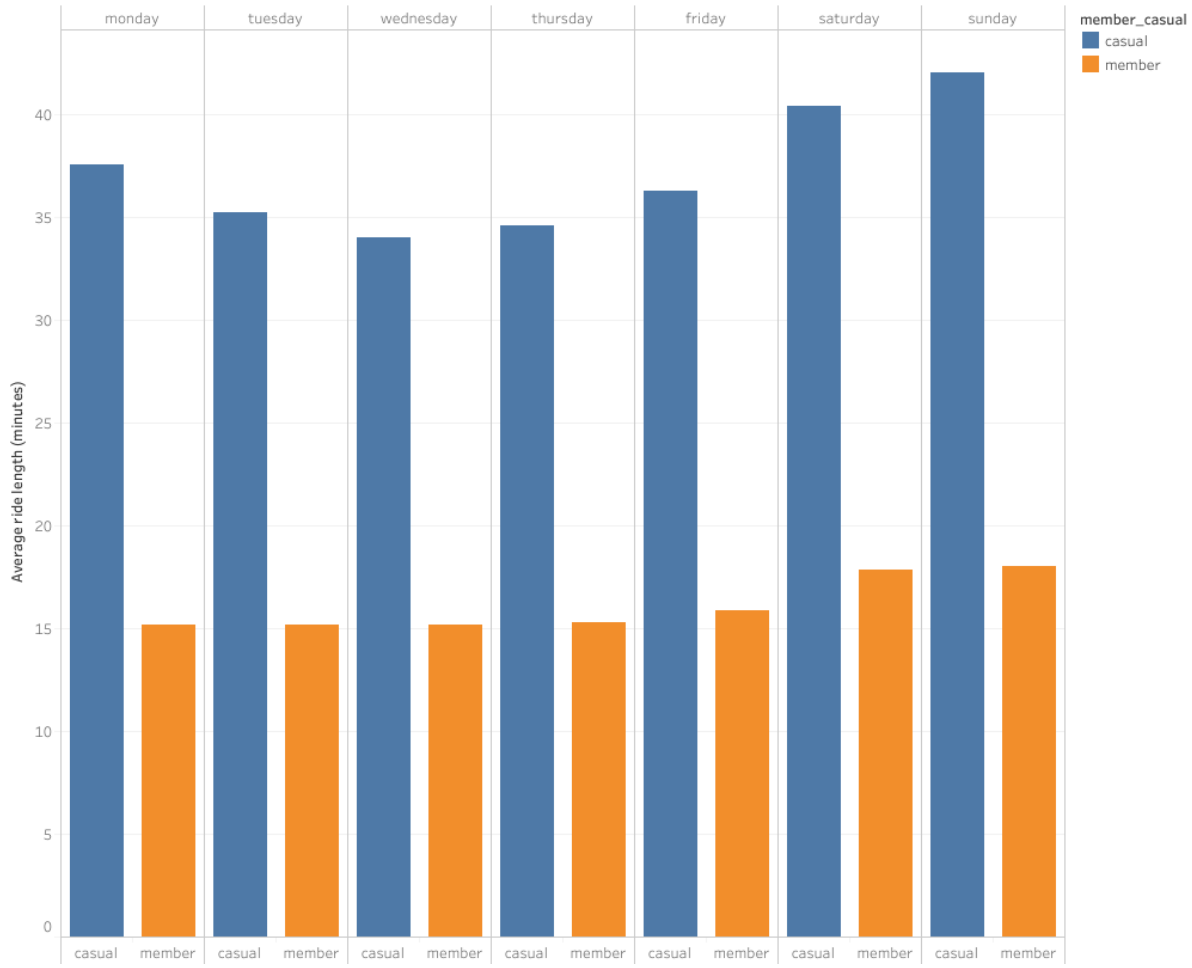
Average ride length for members and users
april 2019 to april 2020

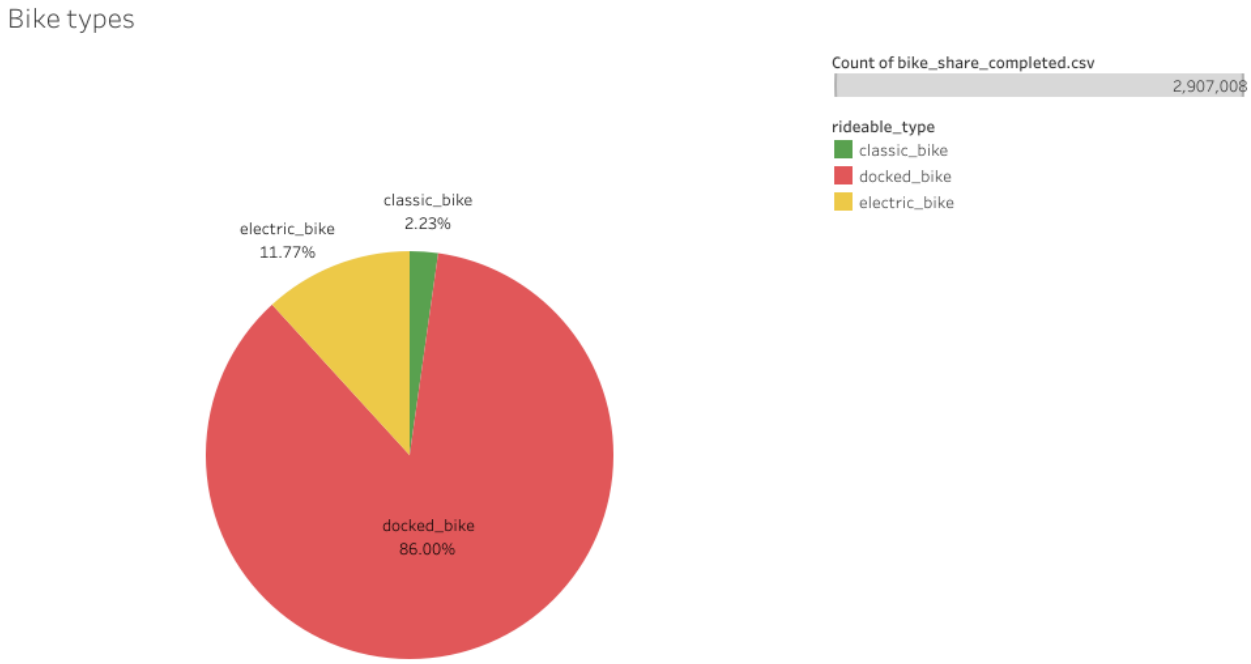


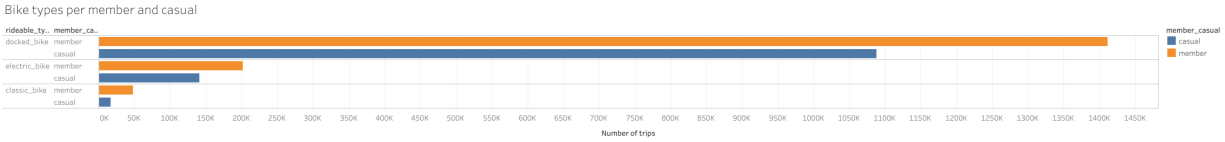
Average ride length for members and casuals per month



Average ride length in minutes per day of the week







Act

Key findings

Bike trips

* Casual riders make up 43% of the total amount of trips taken as opposed to 57% for members.
* Both casual and members show the same trend of bike trips throughout the year. Peaking between June and October.
* Member bike trips stay up a month longer to November before experiencing the same steep drop off.
* Casual riders primarily take trips on the weekends. Members take trips more evenly spread out over the week.
* Most bike trips for casual riders start between 12.00 and 18.00. The starting time for members shows a morning peak between 6.00 and 9.00.

Ride length

* Casual riders (37 minutes) use their bikes 2.4 times longer than members. (16 minutes)
* Casual rider ride length peaks between june and october. Members maintain a more steady ride length throughout the year.
* Ride length increases on Friday, Saturday and Sunday for casual riders. For members the average trip length does not vary much per week day.

Bike types
* Docked bikes are by far the most used bike type by both members and casual riders.
* The classical bike is used significantly less by casual riders than members.

Bussiness statement:

How do annual members and casual riders use rental bikes differently?

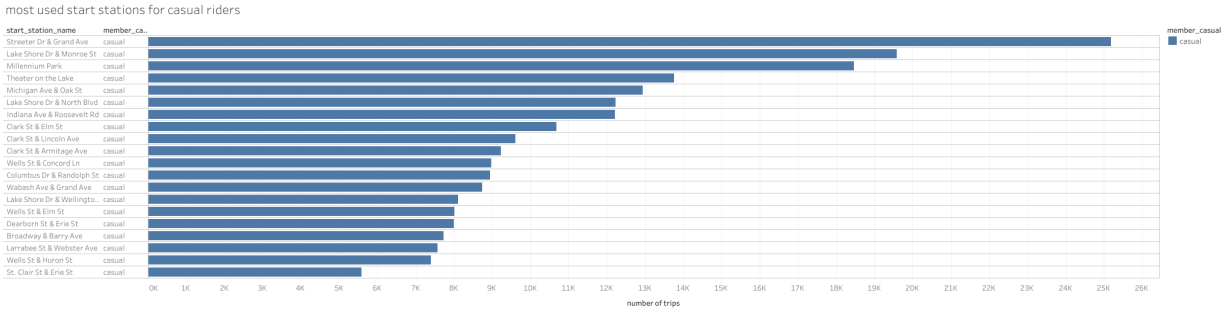
The data shows that casual riders primarily take bike trips during the weekend as opposed to members who take bike trips more evenly spread throughout the week. Casual riders on average also take 2.4 times longer for a single trip, starting their trips later in the day. Both casuals and members take bike trips primarily during the warmer months with a steep decline during the colder months of the year.

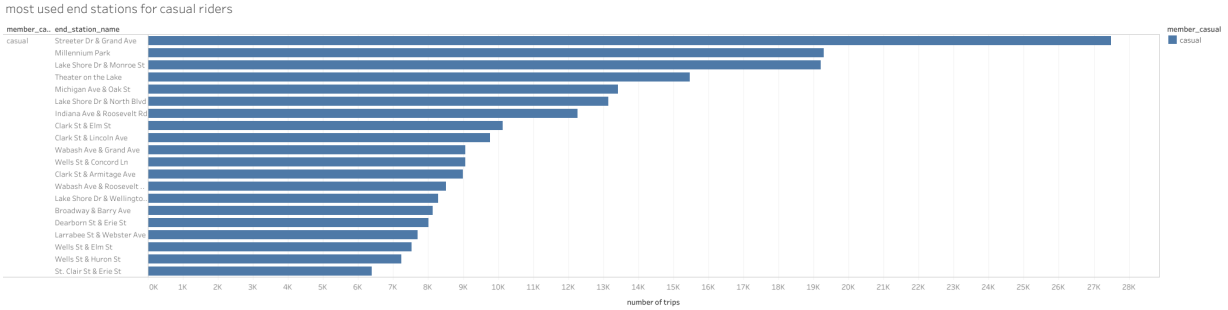
We can therefore conclude that casuals riders on average use the Cyclistic bike services primarily for leisure and not to commute from and to work. At the moment Cyclistic offers a single annual membership which does not benefit casual riders as they primarily take trips on the weekends and during the warmer months. My top 3 recommendations therefore are designed to better fit the needs of casual riders.

Top 3 recommendations

- 1. Offer a **weekend-only membership** at a different price point than the full annual membership to entice casual users towards a full annual membership that is valid from Fridays to Sundays.
- 2. Offer a **half year only_membership** from May to October instead of the full year annual membership.
- 3. Combining the above described recommendations, a third option would be to create a half_year_only membership that is only valid on Friday to Sunday.

To the marketing department:
Below I've included a list of the top 20 most used start and end stations, as well as a list with the most popular routes with the average trip length for each station. You can also get full acces to the file here: Link to the slide show on tableau





most popular routes for casuals