Cyclistic and Data Visualization: “Advanced, Straightforward, and Peeled” (Case Study)

Praveen Choragudi

2022-10-24

Cyclistic Bikes Full Year Analysis from Q4 of 2021 to Q3 of 2022

Based on Kevin Hartman’s “‘Sophisticated, Clear, and Polished’: Divvy and Data Visualization” Divvy case study, which can be found at https://artscience.blog/home/divvy-dataviz-case-study (https://artscience.blog/home/divvy-dataviz-case-study), this analysis. This script’s goal is to compile the Cyclistic data that has been obtained into a single dataframe and then perform a quick analysis to shed light on the fundamental question: “How do members and casual riders use Cyclistic bikes differently?”

Welcome to the case study on Cyclistic’s bike sharing programme! which is a fictitious business. We will use the steps of the data analysis process —ask, prepare, process, analyse, communicate, and act—to provide answers to the important business issues. You may keep on track by using the Case Study Roadmap tables, which include directional questions and important tasks.

Install required packages * tidyverse for data import and wrangling * lubridate for date functions * ggplot for visualization

```
library(tidyverse)  #helps wrangle data
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.5
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(lubridate)  #helps wrangle date attributes
```

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

```
library(ggplot2)  #helps visualize data
getwd() #displays your working directory
```

```
## [1] "/Users/praveenchoragudi/Desktop/Cyclistic_Bikes"
```

Preparing quarterly data by merging multiple csv files

preparing file for Q4 of 2021

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
setwd("~/Desktop/Cyclistic_Bikes/Cyclistic_Bike_Share/Data/Trips/2021")
files<-list.files(pattern = ".csv")
temp<-lapply(files,fread,sep=",")
data<-rbindlist(temp)
write.csv(data,file="Cyclistic_Trips_2021_Q4.csv",row.names = FALSE)
```

preparing file for Q1 of 2022

```
library(data.table)
setwd("~/Desktop/Cyclistic_Bikes/Cyclistic_Bike_Share/Data/Trips/2022/Q1")
files<-list.files(pattern = ".csv")
temp<-lapply(files,fread,sep=",")
data<-rbindlist(temp)
write.csv(data,file="Cyclistic_Trips_2022_Q1.csv",row.names = FALSE)
```

preparing file for Q2 of 2022

```
library(data.table)
setwd("~/Desktop/Cyclistic_Bikes/Cyclistic_Bike_Share/Data/Trips/2022/Q2")
files<-list.files(pattern = ".csv")
temp<-lapply(files,fread,sep=",")
data<-rbindlist(temp)
write.csv(data,file="Cyclistic_Trips_2022_Q2.csv",row.names = FALSE)
```

preparing file for Q3 of 2022

```
library(data.table)
setwd("~/Desktop/Cyclistic_Bikes/Cyclistic_Bike_Share/Data/Trips/2022/Q3")
files<-list.files(pattern = ".csv")
temp<-lapply(files,fread,sep=",")
data<-rbindlist(temp)
write.csv(data,file="Cyclistic_Trips_2022_Q3.csv",row.names = FALSE)
```

STEP 1: COLLECT DATA

```
# Upload Cyclistic datasets (csv files) here

q4_2021 <- read_csv("Cyclistic_Trips_2021_Q4.csv")
```

```
## Rows: 1238744 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (8): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (5): start_lat, start_lng, end_lat, end_lng, day_of_week
## dttm (2): started_at, ended_at
## 
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
q1_2022 <- read_csv("Cyclistic_Trips_2022_Q1.csv")
```

```
## Rows: 503421 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (8): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (5): start_lat, start_lng, end_lat, end_lng, day_of_week
## dttm (2): started_at, ended_at
## 
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
q2_2022 <- read_csv("Cyclistic_Trips_2022_Q2.csv")
```

```
## Rows: 1775311 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (8): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (5): start_lat, start_lng, end_lat, end_lng, day_of_week
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
q3_2022 <- read_csv("Cyclistic_Trips_2022_Q3.csv")
```

```
## Rows: 2310759 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (8): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (5): start_lat, start_lng, end_lat, end_lng, day_of_week
## dttm (2): started_at, ended_at
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

STEP 2: WRANGLING DATA AND COMBINING INTO A SINGLE FILE

Comparing column names each of the files. While the names don't have to be in the same order, they DO need to match perfectly before we can use a command to join them into one file.

```
colnames(q4_2021)
```

```
##  [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"      "ride_length"        "day_of_week"
```

```
colnames(q1_2022)
```

```
##  [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"      "ride_length"        "day_of_week"
```

```
colnames(q2_2022)
```

```
##  [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"      "ride_length"        "day_of_week"
```

```
colnames(q3_2022)
```

```
##  [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"      "ride_length"        "day_of_week"
```

Inspecting the dataframes and looking for incongruencies

```
str(q4_2021)
```

```
## spec_tbl_df [1,238,744 × 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1238744] "620BC6107255BF4C" "4471C70731AB2E45" "26CA69D43D15EE14" "362947F0437E1
514" ...
##  $ rideable_type     : chr [1:1238744] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct[1:1238744], format: "2021-10-22 12:46:42" "2021-10-21 09:12:37" ...
##  $ ended_at          : POSIXct[1:1238744], format: "2021-10-22 12:49:50" "2021-10-21 09:14:14" ...
##  $ start_station_name: chr [1:1238744] "Kingsbury St & Kinzie St" NA NA NA ...
##  $ start_station_id  : chr [1:1238744] "KA1503000043" NA NA NA ...
##  $ end_station_name  : chr [1:1238744] NA NA NA NA ...
##  $ end_station_id    : chr [1:1238744] NA NA NA NA ...
##  $ start_lat         : num [1:1238744] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:1238744] -87.6 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:1238744] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:1238744] -87.6 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:1238744] "member" "member" "member" "member" ...
##  $ ride_length       : chr [1:1238744] "0:0:03:08" "0:0:01:37" "0:0:07:47" "0:0:01:15" ...
##  $ day_of_week       : num [1:1238744] 6 5 7 7 4 5 5 4 5 4 ...
##  - 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(),
##   ..   ride_length = col_character(),
##   ..   day_of_week = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(q1_2022)
```

```
## spec_tbl_df [503,421 × 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:503421] "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED4191054
06" ...
##  $ rideable_type     : chr [1:503421] "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:503421], format: "2022-01-13 11:59:47" "2022-01-10 08:41:56" ...
##  $ ended_at          : POSIXct[1:503421], format: "2022-01-13 12:02:44" "2022-01-10 08:46:17" ...
##  $ start_station_name: chr [1:503421] "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & F
ullerton Ave" "Clark St & Bryn Mawr Ave" ...
##  $ start_station_id  : chr [1:503421] "525" "525" "TA1306000016" "KA1504000151" ...
##  $ end_station_name  : chr [1:503421] "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave & Fullerton
Ave" "Paulina St & Montrose Ave" ...
##  $ end_station_id    : chr [1:503421] "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...
##  $ start_lat         : num [1:503421] 42 42 41.9 42 41.9 ...
##  $ start_lng         : num [1:503421] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:503421] 42 42 41.9 42 41.9 ...
##  $ end_lng           : num [1:503421] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr [1:503421] "casual" "casual" "member" "casual" ...
##  $ ride_length       : chr [1:503421] "0:0:02:57" "0:0:04:21" "0:0:04:21" "0:0:14:56" ...
##  $ day_of_week       : num [1:503421] 5 2 3 3 5 3 1 7 2 6 ...
##  - 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(),
##   ..   ride_length = col_character(),
##   ..   day_of_week = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(q2_2022)
```

```
## spec_tbl_df [1,775,311 × 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1775311] "3564070EEFD12711" "0B820C7FCF22F489" "89EEEE32293F07FF" "84D4751AEB318
88D" ...
##  $ rideable_type     : chr [1:1775311] "electric_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:1775311], format: "2022-04-06 17:42:48" "2022-04-24 19:23:07" ...
##  $ ended_at          : POSIXct[1:1775311], format: "2022-04-06 17:54:36" "2022-04-24 19:43:17" ...
##  $ start_station_name: chr [1:1775311] "Paulina St & Howard St" "Wentworth Ave & Cermak Rd" "Halsted St & Polk
St" "Wentworth Ave & Cermak Rd" ...
##  $ start_station_id  : chr [1:1775311] "515" "13075" "TA1307000121" "13075" ...
##  $ end_station_name  : chr [1:1775311] "University Library (NU)" "Green St & Madison St" "Green St & Madison S
t" "Delano Ct & Roosevelt Rd" ...
##  $ end_station_id    : chr [1:1775311] "605" "TA1307000120" "TA1307000120" "KA1706005007" ...
##  $ start_lat         : num [1:1775311] 42 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:1775311] -87.7 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:1775311] 42.1 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:1775311] -87.7 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:1775311] "member" "member" "member" "casual" ...
##  $ ride_length       : chr [1:1775311] "0:0:11:48" "0:0:20:10" "0:0:06:08" "0:0:09:23" ...
##  $ day_of_week       : num [1:1775311] 4 1 4 6 7 5 2 3 6 6 ...
##  - 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(),
##   ..   ride_length = col_character(),
##   ..   day_of_week = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(q3_2022)
```

```
## spec_tbl_df [2,310,759 × 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:2310759] "954144C2F67B1932" "292E027607D218B6" "57765852588AD6E0" "B5B6BE4431459
0E6" ...
##  $ rideable_type     : chr [1:2310759] "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:2310759], format: "2022-07-05 08:12:47" "2022-07-26 12:53:38" ...
##  $ ended_at          : POSIXct[1:2310759], format: "2022-07-05 08:24:32" "2022-07-26 12:55:31" ...
##  $ start_station_name: chr [1:2310759] "Ashland Ave & Blackhawk St" "Buckingham Fountain (Temp)" "Buckingham F
ountain (Temp)" "Buckingham Fountain (Temp)" ...
##  $ start_station_id  : chr [1:2310759] "13224" "15541" "15541" "15541" ...
##  $ end_station_name  : chr [1:2310759] "Kingsbury St & Kinzie St" "Michigan Ave & 8th St" "Michigan Ave & 8th
St" "Woodlawn Ave & 55th St" ...
##  $ end_station_id    : chr [1:2310759] "KA1503000043" "623" "623" "TA1307000164" ...
##  $ start_lat         : num [1:2310759] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:2310759] -87.7 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:2310759] 41.9 41.9 41.9 41.8 41.9 ...
##  $ end_lng           : num [1:2310759] -87.6 -87.6 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr [1:2310759] "member" "casual" "casual" "casual" ...
##  $ ride_length       : chr [1:2310759] "0:0:11:45" "0:0:01:53" "0:0:07:43" "0:0:58:29" ...
##  $ day_of_week       : num [1:2310759] 3 3 1 1 4 6 2 5 1 1 ...
##  - 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(),
##   ..   ride_length = col_character(),
##   ..   day_of_week = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Stacking individual quarter’s data frames into one big data frame

```
all_trips <- bind_rows(q4_2021, q1_2022, q2_2022, q3_2022)
```

Removing lat, long, and gender fields as this data was dropped beginning in 2020

```
all_trips <- all_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng))
```

STEP 3: CLEAN UP AND ADD DATA TO PREPARE FOR ANALYSIS

Inspecting the new table that has been created

```
colnames(all_trips)  #List of column names
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "member_casual"
## [10] "ride_length"        "day_of_week"
```

```
nrow(all_trips)  #How many rows are in data frame?
```

```
## [1] 5828235
```

```
dim(all_trips)  #Dimensions of the data frame?
```

```
## [1] 5828235      11
```

```
head(all_trips)  #See the first 6 rows of data frame.  Also tail(all_trips)
```

```
## # A tibble: 6 × 11
##   ride_id        ridea…¹ started_at          ended_at            start…² start…³
##   <chr>          <chr>   <dttm>              <dttm>              <chr>   <chr>
## 1 620BC6107255B… electr… 2021-10-22 12:46:42 2021-10-22 12:49:50 Kingsb… KA1503…
## 2 4471C70731AB2… electr… 2021-10-21 09:12:37 2021-10-21 09:14:14 <NA>    <NA>
## 3 26CA69D43D15E… electr… 2021-10-16 16:28:39 2021-10-16 16:36:26 <NA>    <NA>
## 4 362947F0437E1… electr… 2021-10-16 16:17:48 2021-10-16 16:19:03 <NA>    <NA>
## 5 BB731DE2F2EC5… electr… 2021-10-20 23:17:54 2021-10-20 23:26:10 <NA>    <NA>
## 6 7176307BBC097… electr… 2021-10-21 16:57:37 2021-10-21 17:11:58 <NA>    <NA>
## # … with 5 more variables: end_station_name <chr>, end_station_id <chr>,
## #   member_casual <chr>, ride_length <chr>, day_of_week <dbl>, and abbreviated
## #   variable names ¹rideable_type, ²start_station_name, ³start_station_id
```

```
str(all_trips)  #See list of columns and data types (numeric, character, etc)
```

```
## tibble [5,828,235 × 11] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5828235] "620BC6107255BF4C" "4471C70731AB2E45" "26CA69D43D15EE14" "362947F0437E1
514" ...
##  $ rideable_type     : chr [1:5828235] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct[1:5828235], format: "2021-10-22 12:46:42" "2021-10-21 09:12:37" ...
##  $ ended_at          : POSIXct[1:5828235], format: "2021-10-22 12:49:50" "2021-10-21 09:14:14" ...
##  $ start_station_name: chr [1:5828235] "Kingsbury St & Kinzie St" NA NA NA ...
##  $ start_station_id  : chr [1:5828235] "KA1503000043" NA NA NA ...
##  $ end_station_name  : chr [1:5828235] NA NA NA NA ...
##  $ end_station_id    : chr [1:5828235] NA NA NA NA ...
##  $ member_casual     : chr [1:5828235] "member" "member" "member" "member" ...
##  $ ride_length       : chr [1:5828235] "0:0:03:08" "0:0:01:37" "0:0:07:47" "0:0:01:15" ...
##  $ day_of_week       : num [1:5828235] 6 5 7 7 4 5 5 4 5 4 ...
```

```
summary(all_trips)  #Statistical summary of data. Mainly for numerics
```

```
##    ride_id          rideable_type        started_at                   
##  Length:5828235     Length:5828235     Min.   :2021-10-01 00:00:09.00  
##  Class :character   Class :character   1st Qu.:2022-02-28 19:21:08.50  
##  Mode  :character   Mode  :character   Median :2022-06-08 06:41:28.00  
##                                        Mean   :2022-05-06 21:39:18.18  
##                                        3rd Qu.:2022-08-02 11:26:01.00  
##                                        Max.   :2022-09-30 23:59:56.00  
##     ended_at                     start_station_name start_station_id  
##  Min.   :2021-10-01 00:03:11.0   Length:5828235     Length:5828235    
##  1st Qu.:2022-02-28 19:34:02.5   Class :character   Class :character  
##  Median :2022-06-08 06:55:07.0   Mode  :character   Mode  :character  
##  Mean   :2022-05-06 21:58:54.2                                        
##  3rd Qu.:2022-08-02 11:46:26.0                                        
##  Max.   :2022-10-05 19:53:11.0                                        
##  end_station_name   end_station_id     member_casual      ride_length       
##  Length:5828235     Length:5828235     Length:5828235     Length:5828235    
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##   day_of_week   
##  Min.   :1.000  
##  1st Qu.:2.000  
##  Median :4.000  
##  Mean   :4.117  
##  3rd Qu.:6.000  
##  Max.   :7.000
```

Adding columns that list the date, month, day, and year of each ride which allows us to aggregate ride data for each month, day, or year before completing these operations we could only aggregate at the ride level more on date formats in R found at that link (https://www.statmethods.net/input/dates.html).

```
all_trips$date <- as.Date(all_trips$started_at) #The default format is yyyy-mm-dd
all_trips$month <- format(as.Date(all_trips$date), "%m")
all_trips$day <- format(as.Date(all_trips$date), "%d")
all_trips$year <- format(as.Date(all_trips$date), "%Y")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")
```

Adding a “ride_length” calculation (https://stat.ethz.ch/R-manual/R-devel/library/base/html/difftime.html) to all_trips (in seconds)

```
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)
```

Inspecting the structure of the columns

```
str(all_trips)
```

```
## tibble [5,828,235 × 15] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5828235] "620BC6107255BF4C" "4471C70731AB2E45" "26CA69D43D15EE14" "362947F0437E1
514" ...
##  $ rideable_type     : chr [1:5828235] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct[1:5828235], format: "2021-10-22 12:46:42" "2021-10-21 09:12:37" ...
##  $ ended_at          : POSIXct[1:5828235], format: "2021-10-22 12:49:50" "2021-10-21 09:14:14" ...
##  $ start_station_name: chr [1:5828235] "Kingsbury St & Kinzie St" NA NA NA ...
##  $ start_station_id  : chr [1:5828235] "KA1503000043" NA NA NA ...
##  $ end_station_name  : chr [1:5828235] NA NA NA NA ...
##  $ end_station_id    : chr [1:5828235] NA NA NA NA ...
##  $ member_casual     : chr [1:5828235] "member" "member" "member" "member" ...
##  $ ride_length       : 'difftime' num [1:5828235] 188 97 467 75 ...
##   ..- attr(*, "units")= chr "secs"
##  $ day_of_week       : chr [1:5828235] "Friday" "Thursday" "Saturday" "Saturday" ...
##  $ date              : Date[1:5828235], format: "2021-10-22" "2021-10-21" ...
##  $ month             : chr [1:5828235] "10" "10" "10" "10" ...
##  $ day               : chr [1:5828235] "22" "21" "16" "16" ...
##  $ year              : chr [1:5828235] "2021" "2021" "2021" "2021" ...
```

Converting “ride_length” from Factor to numeric so we can run calculations on the data

```
is.factor(all_trips$ride_length)
```

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

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

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

Removing “bad” data. The dataframe (https://www.datasciencemadesimple.com/delete-or-drop-rows-in-r-with-conditions-2/) includes a few hundred entries when bikes were taken out of docks and checked for quality by Cyclistic or ride_length was negative. We will create a new version of the dataframe (v2) since data is being removed.

```
all_trips_v2 <- all_trips[!(all_trips$ride_length<=0),]
```

STEP 4: CONDUCT DESCRIPTIVE ANALYSIS

Descriptive analysis on ride_length (all figures in seconds)

```
mean(all_trips_v2$ride_length) #straight average (total ride length / rides)
```

```
## [1] 1176.375
```

```
median(all_trips_v2$ride_length) #midpoint number in the ascending array of ride lengths
```

```
## [1] 629
```

```
max(all_trips_v2$ride_length) #longest ride
```

```
## [1] 2442301
```

```
min(all_trips_v2$ride_length) #shortest ride
```

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

We can condense the four lines above to one line using summary() on the specific attribute

```
summary(all_trips_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       1     356     629    1176    1131 2442301
```

Comparing members and casual users

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                1761.8174
## 2                     member                 766.1685
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = median)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                      807
## 2                     member                      533
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = max)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                  2442301
## 2                     member                    93594
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = min)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                        1
## 2                     member                        1
```

We can see the average ride time by each day for members vs casual users

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Friday                1680.8608
## 2                      member                   Friday                 751.7498
## 3                      casual                   Monday                1783.8471
## 4                      member                   Monday                 739.7066
## 5                      casual                 Saturday                1962.7752
## 6                      member                 Saturday                 855.8823
## 7                      casual                   Sunday                2062.0366
## 8                      member                   Sunday                 852.9411
## 9                      casual                 Thursday                1540.9259
## 10                     member                 Thursday                 737.6950
## 11                     casual                  Tuesday                1548.6993
## 12                     member                  Tuesday                 729.8295
## 13                     casual                Wednesday                1502.1538
## 14                     member                Wednesday                 727.4074
```

Notice that the days of the week are out of order. Let’s fix that.

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday"
, "Thursday", "Friday", "Saturday"))
```

Now, let’s run the average ride time by each day for members vs casual users

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Sunday                2062.0366
## 2                      member                   Sunday                 852.9411
## 3                      casual                   Monday                1783.8471
## 4                      member                   Monday                 739.7066
## 5                      casual                  Tuesday                1548.6993
## 6                      member                  Tuesday                 729.8295
## 7                      casual                Wednesday                1502.1538
## 8                      member                Wednesday                 727.4074
## 9                      casual                 Thursday                1540.9259
## 10                     member                 Thursday                 737.6950
## 11                     casual                   Friday                1680.8608
## 12                     member                   Friday                 751.7498
## 13                     casual                 Saturday                1962.7752
## 14                     member                 Saturday                 855.8823
```

```
str(all_trips_v2)
```

```
## tibble [5,827,664 × 15] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5827664] "620BC6107255BF4C" "4471C70731AB2E45" "26CA69D43D15EE14" "362947F0437E1
514" ...
##  $ rideable_type     : chr [1:5827664] "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : POSIXct[1:5827664], format: "2021-10-22 12:46:42" "2021-10-21 09:12:37" ...
##  $ ended_at          : POSIXct[1:5827664], format: "2021-10-22 12:49:50" "2021-10-21 09:14:14" ...
##  $ start_station_name: chr [1:5827664] "Kingsbury St & Kinzie St" NA NA NA ...
##  $ start_station_id  : chr [1:5827664] "KA1503000043" NA NA NA ...
##  $ end_station_name  : chr [1:5827664] NA NA NA NA ...
##  $ end_station_id    : chr [1:5827664] NA NA NA NA ...
##  $ member_casual     : chr [1:5827664] "member" "member" "member" "member" ...
##  $ ride_length       : num [1:5827664] 188 97 467 75 496 861 161 501 448 509 ...
##  $ day_of_week       : Ord.factor w/ 7 levels "Sunday"<"Monday"<..: 6 5 7 7 4 5 5 4 5 4 ...
##  $ date              : Date[1:5827664], format: "2021-10-22" "2021-10-21" ...
##  $ month             : chr [1:5827664] "10" "10" "10" "10" ...
##  $ day               : chr [1:5827664] "22" "21" "16" "16" ...
##  $ year              : chr [1:5827664] "2021" "2021" "2021" "2021" ...
```

Analyzing ridership data by type and weekday

```
all_trips_v2 %>%
  group_by(member_casual, day_of_week) %>%                #groups by usertype and weekday
  summarise(number_of_rides = n()                         #calculates the number of rides and average duration
            ,average_duration = mean(ride_length)) %>%    # calculates the average duration
  arrange(member_casual, day_of_week)                     # sorts
```

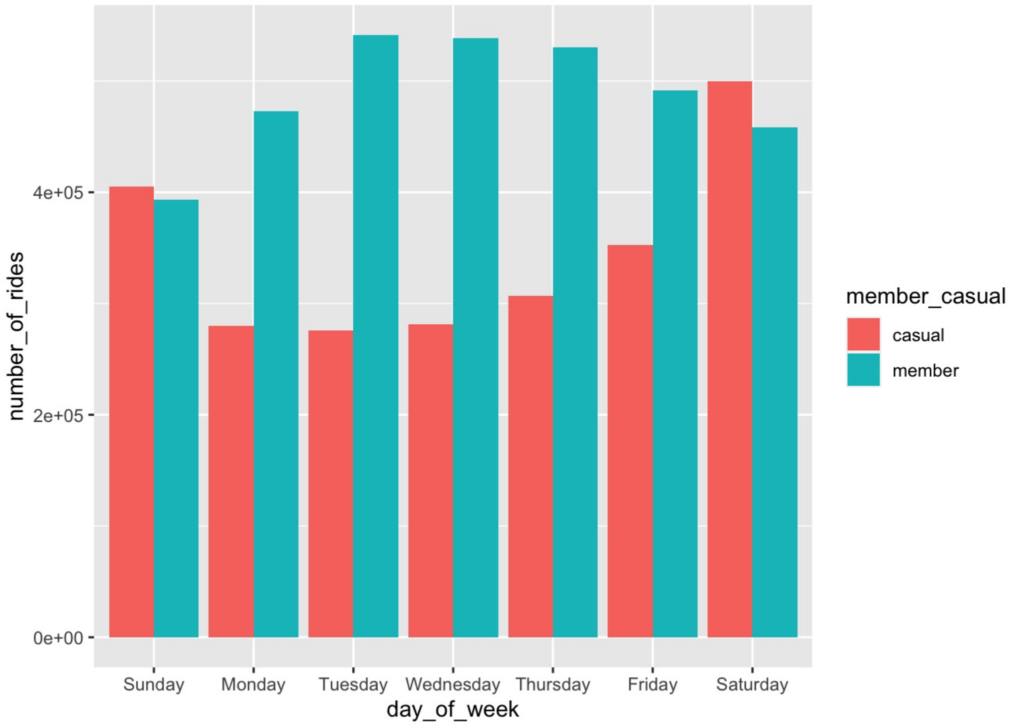
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 14 × 4
## # Groups:   member_casual [2]
##    member_casual day_of_week number_of_rides average_duration
##    <chr>         <ord>                 <int>            <dbl>
##  1 casual        Sunday               404977            2062.
##  2 casual        Monday               279762            1784.
##  3 casual        Tuesday              275745            1549.
##  4 casual        Wednesday            281640            1502.
##  5 casual        Thursday             306662            1541.
##  6 casual        Friday               352466            1681.
##  7 casual        Saturday             499739            1963.
##  8 member        Sunday               393568             853.
##  9 member        Monday               473027             740.
## 10 member        Tuesday              541484             730.
## 11 member        Wednesday            538459             727.
## 12 member        Thursday             530510             738.
## 13 member        Friday               491436             752.
## 14 member        Saturday             458189             856.
```

Let’s visualize the number of rides by rider type

```
all_trips_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

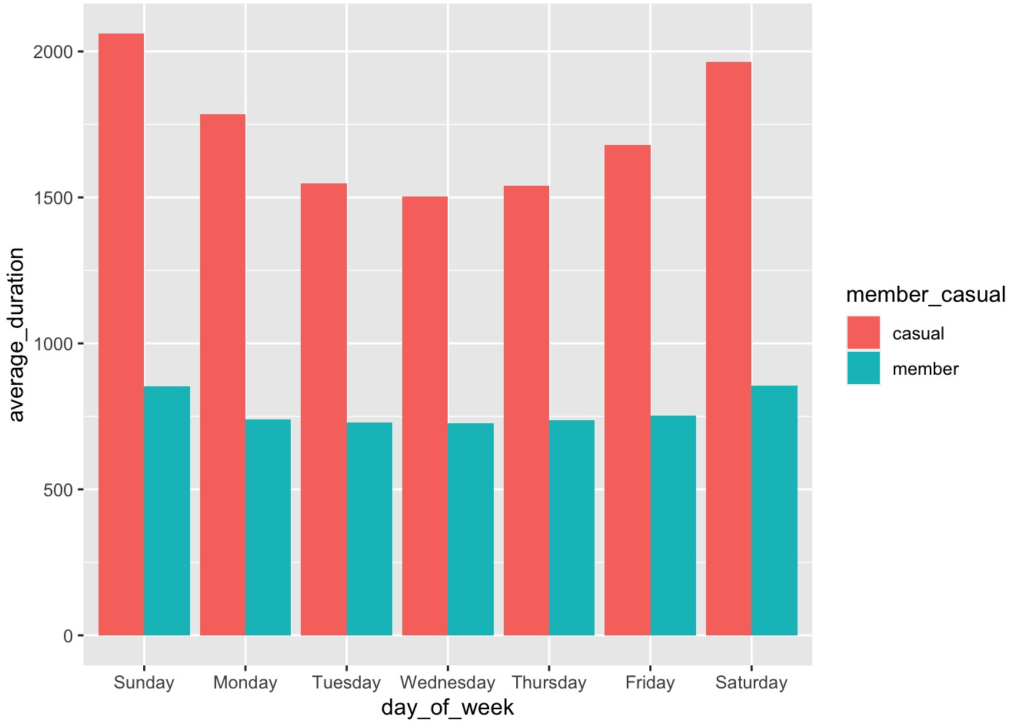
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



Let's create a visualization for average duration

```
all_trips_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

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



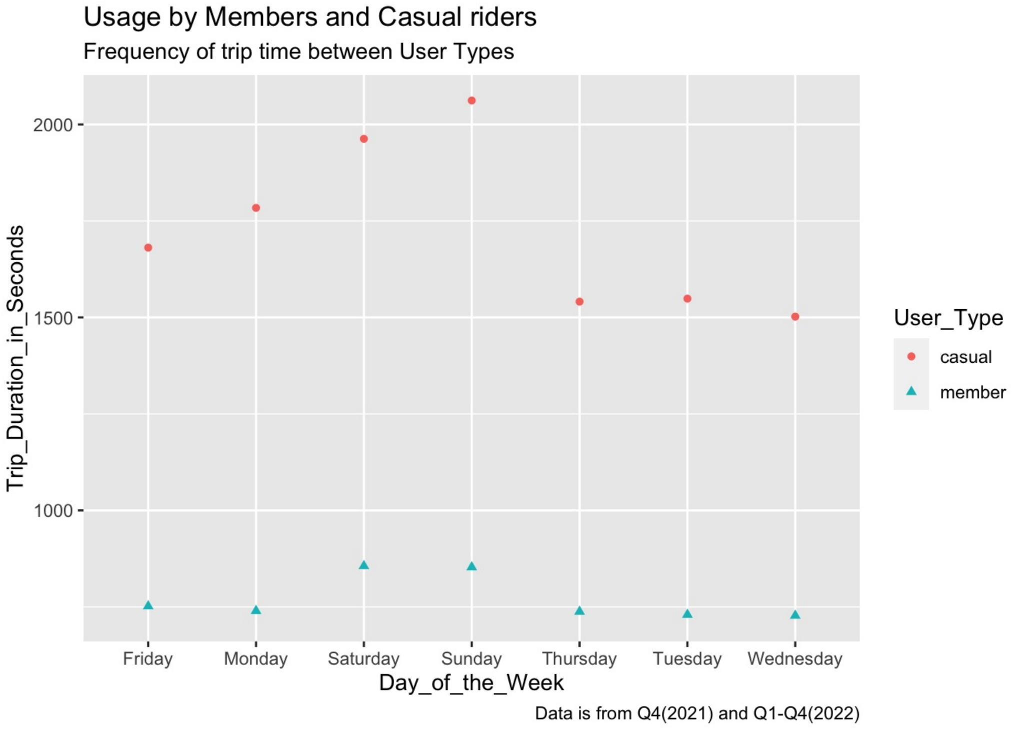
STEP 5: EXPORT SUMMARY FILE FOR FURTHER ANALYSIS

```
counts <- aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
write.csv(counts, file = '~/Desktop/Cyclistic_Bikes/avg_ride_length.csv')

chart1<-read.csv("avg_ride_length.csv")
colnames(chart1)<-c("Count","User_Type","Day_of_the_Week","Trip_Duration_in_Seconds")
colnames(chart1)
```

```
## [1] "Count"                    "User_Type"
## [3] "Day_of_the_Week"          "Trip_Duration_in_Seconds"
```

```
library(ggplot2)
ggplot(data=chart1)+geom_point(mapping = aes(x=Day_of_the_Week,y=Trip_Duration_in_Seconds, color=User_Type,shape=
User_Type))+labs(title ="Usage by Members and Casual riders" ,subtitle ="Frequency of trip time between User Type
s",caption = "Data is from Q4(2021) and Q1-Q4(2022)")
```


Usage by Members and Casual riders

Frequency of trip time between User Types

Data is from Q4(2021) and Q1-Q4(2022)

THANK YOU