Cyclistic Bike Sharing in the Windy City – Chicago, IL

Darryl Nichols

01/03/2022

## The Case Study Scenario as provided by Google/Coursera

“You are a data analyst working 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, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations.”

## Phase 1: Ask Questions to Make Data-Driven Decisions

Three questions will guide the future marketing program: 1. How do annual members and casual riders use Cyclistic bikes differently? 2. Why would casual riders buy Cyclistic annual memberships? 3. How can Cyclistic use digital media to influence casual riders to become members?

I have been assigned the first question to answer.

Action Steps: Collaborate with stakeholders to define the business problem, establish communication preferences, view context of the problem, and establish expectations. Agree on Scope of Work.

Ask SMART Questions to develop business task and Scope of Work:

* Specific
* Measurable
* Action-Oriented
* Relevant
* Time-Bound

Business Task: How do annual members and casual riders use Cyclistic bikes differently?

## Phase 2: Prepare Data for Exploration

Action steps: Decide what data is necessary to address business task, locate the data, create any security measures to protect the data, and decide on key metrics to use when completing the business task.

Does the data ROCCC? Is the data…

* Reliable - At a glance, our data seems to unbiased and complete.
* Original - We are assuming that we (Cyclistic) have collected our own data and is First Party data.
* Comprehensive - The data contains information we need to answer the business question.
* Current- Yes. The data is from December 2021.
* Cited - This data is made available from Motivate International
* Download data and store it appropriately - Files were originally contained in zip files, then saved as .csv files. See *Collect Data*
* Identity how the data is organized - Data is organized into long data, observe data types and metadata, structure. See *Preview Data*
* Sort and filter data - there are many 00:00:00 (HH:MM:SS) values as well as negative values - more on the implications of this in the analyze phase.
* Determine the credibility of the data - as outlined above, this data ROCCC’s.

#### Install required packages/Load Libraries

install.packages(“tidyverse”) install.packages(“lubridate”) install.packages(“ggplot”) install.packages(“skimr”) install.packages(“janitor”)

getwd() #displays the working directory setwd(…) #sets the working directory to simplify calls to data

library(tidyverse) # helps transform/clean data

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.1.1 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(lubridate) # helps wrangle/parse date attributes

##   
## Attaching package: 'lubridate'

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

library(ggplot2) # helps visualize data  
library(readr) #  
library(skimr) #  
library(janitor) #

##   
## Attaching package: 'janitor'

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

library(dplyr) #

#### COLLECT DATA

Upload Divvy datasets (csv files) here from the tidyverse package and readr library

q2\_2019 <- read\_csv("Divvy\_Trips\_2019\_Q2.csv")

## Rows: 1108163 Columns: 12

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (4): 03 - Rental Start Station Name, 02 - Rental End Station Name, User...  
## dbl (5): 01 - Rental Details Rental ID, 01 - Rental Details Bike ID, 03 - R...  
## dttm (2): 01 - Rental Details Local Start Time, 01 - Rental Details Local En...

##   
## 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\_2019 <- read\_csv("Divvy\_Trips\_2019\_Q3.csv")

## Rows: 1640718 Columns: 12

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (4): from\_station\_name, to\_station\_name, usertype, gender  
## dbl (5): trip\_id, bikeid, from\_station\_id, to\_station\_id, birthyear  
## dttm (2): start\_time, end\_time

##   
## 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.

q4\_2019 <- read\_csv("Divvy\_Trips\_2019\_Q4.csv")

## Rows: 704054 Columns: 12

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (4): from\_station\_name, to\_station\_name, usertype, gender  
## dbl (5): trip\_id, bikeid, from\_station\_id, to\_station\_id, birthyear  
## dttm (2): start\_time, end\_time

##   
## 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\_2020 <- read\_csv("Divvy\_Trips\_2020\_Q1.csv")

## Rows: 426887 Columns: 13

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (5): ride\_id, rideable\_type, start\_station\_name, end\_station\_name, memb...  
## dbl (6): start\_station\_id, end\_station\_id, start\_lat, start\_lng, end\_lat, e...  
## 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.

#### PREVIEW DATAFRAMES

Replace the file and column names as desired to preview all 4 dataframes.

skim\_without\_charts(q3\_2019) #dyplr

Data summary

|  |  |
| --- | --- |
| Name | q3\_2019 |
| Number of rows | 1640718 |
| Number of columns | 12 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 4 |
| numeric | 6 |
| POSIXct | 2 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| from\_station\_name | 0 | 1.00 | 10 | 43 | 0 | 612 | 0 |
| to\_station\_name | 0 | 1.00 | 10 | 43 | 0 | 613 | 0 |
| usertype | 0 | 1.00 | 8 | 10 | 0 | 2 | 0 |
| gender | 287350 | 0.82 | 4 | 6 | 0 | 2 | 0 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| trip\_id | 0 | 1.00 | 24364471.07 | 499548.45 | 23479388 | 23935498 | 24367416 | 24797401 | 25223639 |
| bikeid | 0 | 1.00 | 3349.86 | 1888.88 | 1 | 1713 | 3419 | 4997 | 6471 |
| tripduration | 0 | 1.00 | 1741.74 | 38503.44 | 61 | 465 | 813 | 1460 | 9056633 |
| from\_station\_id | 0 | 1.00 | 202.40 | 156.72 | 2 | 77 | 174 | 289 | 673 |
| to\_station\_id | 0 | 1.00 | 203.90 | 156.70 | 2 | 80 | 176 | 291 | 673 |
| birthyear | 278094 | 0.83 | 1984.90 | 10.61 | 1888 | 1980 | 1988 | 1992 | 2003 |

**Variable type: POSIXct**

| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| start\_time | 0 | 1 | 2019-07-01 00:00:27 | 2019-09-30 23:59:37 | 2019-08-14 07:11:50 | 1372358 |
| end\_time | 0 | 1 | 2019-07-01 00:07:31 | 2019-11-04 08:09:47 | 2019-08-14 07:28:07 | 1344539 |

glimpse(q3\_2019) #dyplr

## Rows: 1,640,718  
## Columns: 12  
## $ trip\_id <dbl> 23479388, 23479389, 23479390, 23479391, 23479392, 23~  
## $ start\_time <dttm> 2019-07-01 00:00:27, 2019-07-01 00:01:16, 2019-07-0~  
## $ end\_time <dttm> 2019-07-01 00:20:41, 2019-07-01 00:18:44, 2019-07-0~  
## $ bikeid <dbl> 3591, 5353, 6180, 5540, 6014, 4941, 3770, 5442, 2957~  
## $ tripduration <dbl> 1214, 1048, 1554, 1503, 1213, 310, 1248, 1550, 1583,~  
## $ from\_station\_id <dbl> 117, 381, 313, 313, 168, 300, 168, 313, 43, 43, 511,~  
## $ from\_station\_name <chr> "Wilton Ave & Belmont Ave", "Western Ave & Monroe St~  
## $ to\_station\_id <dbl> 497, 203, 144, 144, 62, 232, 62, 144, 195, 195, 84, ~  
## $ to\_station\_name <chr> "Kimball Ave & Belmont Ave", "Western Ave & 21st St"~  
## $ usertype <chr> "Subscriber", "Customer", "Customer", "Customer", "C~  
## $ gender <chr> "Male", NA, NA, NA, NA, "Male", NA, NA, NA, NA, NA, ~  
## $ birthyear <dbl> 1992, NA, NA, NA, NA, 1990, NA, NA, NA, NA, NA, NA, ~

head(q3\_2019)

## # A tibble: 6 x 12  
## trip\_id start\_time end\_time bikeid tripduration  
## <dbl> <dttm> <dttm> <dbl> <dbl>  
## 1 23479388 2019-07-01 00:00:27 2019-07-01 00:20:41 3591 1214  
## 2 23479389 2019-07-01 00:01:16 2019-07-01 00:18:44 5353 1048  
## 3 23479390 2019-07-01 00:01:48 2019-07-01 00:27:42 6180 1554  
## 4 23479391 2019-07-01 00:02:07 2019-07-01 00:27:10 5540 1503  
## 5 23479392 2019-07-01 00:02:13 2019-07-01 00:22:26 6014 1213  
## 6 23479393 2019-07-01 00:02:21 2019-07-01 00:07:31 4941 310  
## # ... with 7 more variables: from\_station\_id <dbl>, from\_station\_name <chr>,  
## # to\_station\_id <dbl>, to\_station\_name <chr>, usertype <chr>, gender <chr>,  
## # birthyear <dbl>

q3\_2019 %>%  
 select(trip\_id) # Only view the specified columns in the dataframe from dyplr

## # A tibble: 1,640,718 x 1  
## trip\_id  
## <dbl>  
## 1 23479388  
## 2 23479389  
## 3 23479390  
## 4 23479391  
## 5 23479392  
## 6 23479393  
## 7 23479394  
## 8 23479395  
## 9 23479396  
## 10 23479397  
## # ... with 1,640,708 more rows

# OR  
tibble(q3\_2019$trip\_id)

## # A tibble: 1,640,718 x 1  
## `q3\_2019$trip\_id`  
## <dbl>  
## 1 23479388  
## 2 23479389  
## 3 23479390  
## 4 23479391  
## 5 23479392  
## 6 23479393  
## 7 23479394  
## 8 23479395  
## 9 23479396  
## 10 23479397  
## # ... with 1,640,708 more rows

## Phase 3: Process Data from Dirty to Clean

Action steps: Decide what tools to use for analysis, ensure data integrity, clean data, document cleaning, and verify that data is ready for analysis.

* Choose tools for cleaning - using R because it can handle large amounts of data in file, comes with useful cleaning packages.
* Check the data for errors - looking for duplicate data, inconsistent data types, incomplete data, and inaccurate/incorrect data
* Transform data - checking for spelling errors, changing the case of text, and remove unnecessary spaces/trim.
* Document cleaning process - documented using R Markdown

#### KEY LIMITATION

q2\_2019, q3\_2019, and q4\_2019 NOT have a "rideable\_type" column to distinguish between "docked\_bike", "electric\_bike, and "classic\_bike". Therefore, in the Tableau report from the analyze/share phase, there will be a dashboard analysis with only q1\_2020 regarding "rideable\_type".

#### PREPARE DATA AND COMBINE INTO A SINGLE FILE

Compare column names each of the files the names need to match perfectly before using the bind\_rows command to join them into one file

colnames(q3\_2019)

## [1] "trip\_id" "start\_time" "end\_time"   
## [4] "bikeid" "tripduration" "from\_station\_id"   
## [7] "from\_station\_name" "to\_station\_id" "to\_station\_name"   
## [10] "usertype" "gender" "birthyear"

colnames(q4\_2019)

## [1] "trip\_id" "start\_time" "end\_time"   
## [4] "bikeid" "tripduration" "from\_station\_id"   
## [7] "from\_station\_name" "to\_station\_id" "to\_station\_name"   
## [10] "usertype" "gender" "birthyear"

colnames(q2\_2019)

## [1] "01 - Rental Details Rental ID"   
## [2] "01 - Rental Details Local Start Time"   
## [3] "01 - Rental Details Local End Time"   
## [4] "01 - Rental Details Bike ID"   
## [5] "01 - Rental Details Duration In Seconds Uncapped"  
## [6] "03 - Rental Start Station ID"   
## [7] "03 - Rental Start Station Name"   
## [8] "02 - Rental End Station ID"   
## [9] "02 - Rental End Station Name"   
## [10] "User Type"   
## [11] "Member Gender"   
## [12] "05 - Member Details Member Birthday Year"

colnames(q1\_2020)

## [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"

#### Rename columns to make them consistent with q1\_2020 (this is the most recent table design) from the tidyverse package and dyplr library

(q4\_2019 <- rename(q4\_2019  
 ,ride\_id = trip\_id  
 ,rideable\_type = bikeid   
 ,started\_at = start\_time   
 ,ended\_at = end\_time   
 ,start\_station\_name = from\_station\_name   
 ,start\_station\_id = from\_station\_id   
 ,end\_station\_name = to\_station\_name   
 ,end\_station\_id = to\_station\_id   
 ,member\_casual = usertype))

## # A tibble: 704,054 x 12  
## ride\_id started\_at ended\_at rideable\_type tripduration  
## <dbl> <dttm> <dttm> <dbl> <dbl>  
## 1 25223640 2019-10-01 00:01:39 2019-10-01 00:17:20 2215 940  
## 2 25223641 2019-10-01 00:02:16 2019-10-01 00:06:34 6328 258  
## 3 25223642 2019-10-01 00:04:32 2019-10-01 00:18:43 3003 850  
## 4 25223643 2019-10-01 00:04:32 2019-10-01 00:43:43 3275 2350  
## 5 25223644 2019-10-01 00:04:34 2019-10-01 00:35:42 5294 1867  
## 6 25223645 2019-10-01 00:04:38 2019-10-01 00:10:51 1891 373  
## 7 25223646 2019-10-01 00:04:52 2019-10-01 00:22:45 1061 1072  
## 8 25223647 2019-10-01 00:04:57 2019-10-01 00:29:16 1274 1458  
## 9 25223648 2019-10-01 00:05:20 2019-10-01 00:29:18 6011 1437  
## 10 25223649 2019-10-01 00:05:20 2019-10-01 02:23:46 2957 8306  
## # ... with 704,044 more rows, and 7 more variables: start\_station\_id <dbl>,  
## # start\_station\_name <chr>, end\_station\_id <dbl>, end\_station\_name <chr>,  
## # member\_casual <chr>, gender <chr>, birthyear <dbl>

(q3\_2019 <- rename(q3\_2019  
 ,ride\_id = trip\_id  
 ,rideable\_type = bikeid   
 ,started\_at = start\_time   
 ,ended\_at = end\_time   
 ,start\_station\_name = from\_station\_name   
 ,start\_station\_id = from\_station\_id   
 ,end\_station\_name = to\_station\_name   
 ,end\_station\_id = to\_station\_id   
 ,member\_casual = usertype))

## # A tibble: 1,640,718 x 12  
## ride\_id started\_at ended\_at rideable\_type tripduration  
## <dbl> <dttm> <dttm> <dbl> <dbl>  
## 1 23479388 2019-07-01 00:00:27 2019-07-01 00:20:41 3591 1214  
## 2 23479389 2019-07-01 00:01:16 2019-07-01 00:18:44 5353 1048  
## 3 23479390 2019-07-01 00:01:48 2019-07-01 00:27:42 6180 1554  
## 4 23479391 2019-07-01 00:02:07 2019-07-01 00:27:10 5540 1503  
## 5 23479392 2019-07-01 00:02:13 2019-07-01 00:22:26 6014 1213  
## 6 23479393 2019-07-01 00:02:21 2019-07-01 00:07:31 4941 310  
## 7 23479394 2019-07-01 00:02:24 2019-07-01 00:23:12 3770 1248  
## 8 23479395 2019-07-01 00:02:26 2019-07-01 00:28:16 5442 1550  
## 9 23479396 2019-07-01 00:02:34 2019-07-01 00:28:57 2957 1583  
## 10 23479397 2019-07-01 00:02:45 2019-07-01 00:29:14 6091 1589  
## # ... with 1,640,708 more rows, and 7 more variables: start\_station\_id <dbl>,  
## # start\_station\_name <chr>, end\_station\_id <dbl>, end\_station\_name <chr>,  
## # member\_casual <chr>, gender <chr>, birthyear <dbl>

(q2\_2019 <- rename(q2\_2019  
 ,ride\_id = "01 - Rental Details Rental ID"  
 ,rideable\_type = "01 - Rental Details Bike ID"   
 ,started\_at = "01 - Rental Details Local Start Time"   
 ,ended\_at = "01 - Rental Details Local End Time"   
 ,start\_station\_name = "03 - Rental Start Station Name"   
 ,start\_station\_id = "03 - Rental Start Station ID"  
 ,end\_station\_name = "02 - Rental End Station Name"   
 ,end\_station\_id = "02 - Rental End Station ID"  
 ,member\_casual = "User Type"))

## # A tibble: 1,108,163 x 12  
## ride\_id started\_at ended\_at rideable\_type  
## <dbl> <dttm> <dttm> <dbl>  
## 1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48 6251  
## 2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30 6226  
## 3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19 5649  
## 4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58 4151  
## 5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13 3270  
## 6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56 3123  
## 7 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41 6418  
## 8 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11 4513  
## 9 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44 3280  
## 10 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39 5534  
## # ... with 1,108,153 more rows, and 8 more variables:  
## # 01 - Rental Details Duration In Seconds Uncapped <dbl>,  
## # start\_station\_id <dbl>, start\_station\_name <chr>, end\_station\_id <dbl>,  
## # end\_station\_name <chr>, member\_casual <chr>, Member Gender <chr>,  
## # 05 - Member Details Member Birthday Year <dbl>

#### Check and clean column names post rename for characters, numbers, and underscores only with clean\_names from the janitor package

clean\_names(q2\_2019)

## # A tibble: 1,108,163 x 12  
## ride\_id started\_at ended\_at rideable\_type  
## <dbl> <dttm> <dttm> <dbl>  
## 1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48 6251  
## 2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30 6226  
## 3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19 5649  
## 4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58 4151  
## 5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13 3270  
## 6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56 3123  
## 7 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41 6418  
## 8 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11 4513  
## 9 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44 3280  
## 10 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39 5534  
## # ... with 1,108,153 more rows, and 8 more variables:  
## # x01\_rental\_details\_duration\_in\_seconds\_uncapped <dbl>,  
## # start\_station\_id <dbl>, start\_station\_name <chr>, end\_station\_id <dbl>,  
## # end\_station\_name <chr>, member\_casual <chr>, member\_gender <chr>,  
## # x05\_member\_details\_member\_birthday\_year <dbl>

clean\_names(q3\_2019)

## # A tibble: 1,640,718 x 12  
## ride\_id started\_at ended\_at rideable\_type tripduration  
## <dbl> <dttm> <dttm> <dbl> <dbl>  
## 1 23479388 2019-07-01 00:00:27 2019-07-01 00:20:41 3591 1214  
## 2 23479389 2019-07-01 00:01:16 2019-07-01 00:18:44 5353 1048  
## 3 23479390 2019-07-01 00:01:48 2019-07-01 00:27:42 6180 1554  
## 4 23479391 2019-07-01 00:02:07 2019-07-01 00:27:10 5540 1503  
## 5 23479392 2019-07-01 00:02:13 2019-07-01 00:22:26 6014 1213  
## 6 23479393 2019-07-01 00:02:21 2019-07-01 00:07:31 4941 310  
## 7 23479394 2019-07-01 00:02:24 2019-07-01 00:23:12 3770 1248  
## 8 23479395 2019-07-01 00:02:26 2019-07-01 00:28:16 5442 1550  
## 9 23479396 2019-07-01 00:02:34 2019-07-01 00:28:57 2957 1583  
## 10 23479397 2019-07-01 00:02:45 2019-07-01 00:29:14 6091 1589  
## # ... with 1,640,708 more rows, and 7 more variables: start\_station\_id <dbl>,  
## # start\_station\_name <chr>, end\_station\_id <dbl>, end\_station\_name <chr>,  
## # member\_casual <chr>, gender <chr>, birthyear <dbl>

clean\_names(q4\_2019)

## # A tibble: 704,054 x 12  
## ride\_id started\_at ended\_at rideable\_type tripduration  
## <dbl> <dttm> <dttm> <dbl> <dbl>  
## 1 25223640 2019-10-01 00:01:39 2019-10-01 00:17:20 2215 940  
## 2 25223641 2019-10-01 00:02:16 2019-10-01 00:06:34 6328 258  
## 3 25223642 2019-10-01 00:04:32 2019-10-01 00:18:43 3003 850  
## 4 25223643 2019-10-01 00:04:32 2019-10-01 00:43:43 3275 2350  
## 5 25223644 2019-10-01 00:04:34 2019-10-01 00:35:42 5294 1867  
## 6 25223645 2019-10-01 00:04:38 2019-10-01 00:10:51 1891 373  
## 7 25223646 2019-10-01 00:04:52 2019-10-01 00:22:45 1061 1072  
## 8 25223647 2019-10-01 00:04:57 2019-10-01 00:29:16 1274 1458  
## 9 25223648 2019-10-01 00:05:20 2019-10-01 00:29:18 6011 1437  
## 10 25223649 2019-10-01 00:05:20 2019-10-01 02:23:46 2957 8306  
## # ... with 704,044 more rows, and 7 more variables: start\_station\_id <dbl>,  
## # start\_station\_name <chr>, end\_station\_id <dbl>, end\_station\_name <chr>,  
## # member\_casual <chr>, gender <chr>, birthyear <dbl>

clean\_names(q1\_2020)

## # A tibble: 426,887 x 13  
## ride\_id rideable\_type started\_at ended\_at   
## <chr> <chr> <dttm> <dttm>   
## 1 EACB19130B0CDA4A docked\_bike 2020-01-21 20:06:59 2020-01-21 20:14:30  
## 2 8FED874C809DC021 docked\_bike 2020-01-30 14:22:39 2020-01-30 14:26:22  
## 3 789F3C21E472CA96 docked\_bike 2020-01-09 19:29:26 2020-01-09 19:32:17  
## 4 C9A388DAC6ABF313 docked\_bike 2020-01-06 16:17:07 2020-01-06 16:25:56  
## 5 943BC3CBECCFD662 docked\_bike 2020-01-30 08:37:16 2020-01-30 08:42:48  
## 6 6D9C8A6938165C11 docked\_bike 2020-01-10 12:33:05 2020-01-10 12:37:54  
## 7 31EB9B8F406D4C82 docked\_bike 2020-01-10 13:07:35 2020-01-10 13:12:24  
## 8 A2B24E3F9C9720E3 docked\_bike 2020-01-10 07:24:53 2020-01-10 07:29:50  
## 9 5E3F01E1441730B7 docked\_bike 2020-01-31 16:37:16 2020-01-31 16:42:11  
## 10 19DC57F7E3140131 docked\_bike 2020-01-31 09:39:17 2020-01-31 09:42:40  
## # ... with 426,877 more rows, and 9 more variables: start\_station\_name <chr>,  
## # start\_station\_id <dbl>, end\_station\_name <chr>, end\_station\_id <dbl>,  
## # start\_lat <dbl>, start\_lng <dbl>, end\_lat <dbl>, end\_lng <dbl>,  
## # member\_casual <chr>

#### Inspect the dataframes and look for incongruencies

str(q1\_2020)

## spec\_tbl\_df [426,887 x 13] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ride\_id : chr [1:426887] "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A388DAC6ABF313" ...  
## $ rideable\_type : chr [1:426887] "docked\_bike" "docked\_bike" "docked\_bike" "docked\_bike" ...  
## $ started\_at : POSIXct[1:426887], format: "2020-01-21 20:06:59" "2020-01-30 14:22:39" ...  
## $ ended\_at : POSIXct[1:426887], format: "2020-01-21 20:14:30" "2020-01-30 14:26:22" ...  
## $ start\_station\_name: chr [1:426887] "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway & Belmont Ave" "Clark St & Randolph St" ...  
## $ start\_station\_id : num [1:426887] 239 234 296 51 66 212 96 96 212 38 ...  
## $ end\_station\_name : chr [1:426887] "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilton Ave & Belmont Ave" "Fairbanks Ct & Grand Ave" ...  
## $ end\_station\_id : num [1:426887] 326 318 117 24 212 96 212 212 96 100 ...  
## $ start\_lat : num [1:426887] 42 42 41.9 41.9 41.9 ...  
## $ start\_lng : num [1:426887] -87.7 -87.7 -87.6 -87.6 -87.6 ...  
## $ end\_lat : num [1:426887] 42 42 41.9 41.9 41.9 ...  
## $ end\_lng : num [1:426887] -87.7 -87.7 -87.7 -87.6 -87.6 ...  
## $ member\_casual : chr [1:426887] "member" "member" "member" "member" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. ride\_id = col\_character(),  
## .. rideable\_type = col\_character(),  
## .. started\_at = col\_datetime(format = ""),  
## .. ended\_at = col\_datetime(format = ""),  
## .. start\_station\_name = col\_character(),  
## .. start\_station\_id = col\_double(),  
## .. end\_station\_name = col\_character(),  
## .. end\_station\_id = col\_double(),  
## .. start\_lat = col\_double(),  
## .. start\_lng = col\_double(),  
## .. end\_lat = col\_double(),  
## .. end\_lng = col\_double(),  
## .. member\_casual = col\_character()  
## .. )  
## - attr(\*, "problems")=<externalptr>

str(q4\_2019)

## spec\_tbl\_df [704,054 x 12] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ride\_id : num [1:704054] 25223640 25223641 25223642 25223643 25223644 ...  
## $ started\_at : POSIXct[1:704054], format: "2019-10-01 00:01:39" "2019-10-01 00:02:16" ...  
## $ ended\_at : POSIXct[1:704054], format: "2019-10-01 00:17:20" "2019-10-01 00:06:34" ...  
## $ rideable\_type : num [1:704054] 2215 6328 3003 3275 5294 ...  
## $ tripduration : num [1:704054] 940 258 850 2350 1867 ...  
## $ start\_station\_id : num [1:704054] 20 19 84 313 210 156 84 156 156 336 ...  
## $ start\_station\_name: chr [1:704054] "Sheffield Ave & Kingsbury St" "Throop (Loomis) St & Taylor St" "Milwaukee Ave & Grand Ave" "Lakeview Ave & Fullerton Pkwy" ...  
## $ end\_station\_id : num [1:704054] 309 241 199 290 382 226 142 463 463 336 ...  
## $ end\_station\_name : chr [1:704054] "Leavitt St & Armitage Ave" "Morgan St & Polk St" "Wabash Ave & Grand Ave" "Kedzie Ave & Palmer Ct" ...  
## $ member\_casual : chr [1:704054] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...  
## $ gender : chr [1:704054] "Male" "Male" "Female" "Male" ...  
## $ birthyear : num [1:704054] 1987 1998 1991 1990 1987 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. trip\_id = col\_double(),  
## .. start\_time = col\_datetime(format = ""),  
## .. end\_time = col\_datetime(format = ""),  
## .. bikeid = col\_double(),  
## .. tripduration = col\_number(),  
## .. from\_station\_id = col\_double(),  
## .. from\_station\_name = col\_character(),  
## .. to\_station\_id = col\_double(),  
## .. to\_station\_name = col\_character(),  
## .. usertype = col\_character(),  
## .. gender = col\_character(),  
## .. birthyear = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

str(q3\_2019)

## spec\_tbl\_df [1,640,718 x 12] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ride\_id : num [1:1640718] 23479388 23479389 23479390 23479391 23479392 ...  
## $ started\_at : POSIXct[1:1640718], format: "2019-07-01 00:00:27" "2019-07-01 00:01:16" ...  
## $ ended\_at : POSIXct[1:1640718], format: "2019-07-01 00:20:41" "2019-07-01 00:18:44" ...  
## $ rideable\_type : num [1:1640718] 3591 5353 6180 5540 6014 ...  
## $ tripduration : num [1:1640718] 1214 1048 1554 1503 1213 ...  
## $ start\_station\_id : num [1:1640718] 117 381 313 313 168 300 168 313 43 43 ...  
## $ start\_station\_name: chr [1:1640718] "Wilton Ave & Belmont Ave" "Western Ave & Monroe St" "Lakeview Ave & Fullerton Pkwy" "Lakeview Ave & Fullerton Pkwy" ...  
## $ end\_station\_id : num [1:1640718] 497 203 144 144 62 232 62 144 195 195 ...  
## $ end\_station\_name : chr [1:1640718] "Kimball Ave & Belmont Ave" "Western Ave & 21st St" "Larrabee St & Webster Ave" "Larrabee St & Webster Ave" ...  
## $ member\_casual : chr [1:1640718] "Subscriber" "Customer" "Customer" "Customer" ...  
## $ gender : chr [1:1640718] "Male" NA NA NA ...  
## $ birthyear : num [1:1640718] 1992 NA NA NA NA ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. trip\_id = col\_double(),  
## .. start\_time = col\_datetime(format = ""),  
## .. end\_time = col\_datetime(format = ""),  
## .. bikeid = col\_double(),  
## .. tripduration = col\_number(),  
## .. from\_station\_id = col\_double(),  
## .. from\_station\_name = col\_character(),  
## .. to\_station\_id = col\_double(),  
## .. to\_station\_name = col\_character(),  
## .. usertype = col\_character(),  
## .. gender = col\_character(),  
## .. birthyear = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

str(q2\_2019)

## spec\_tbl\_df [1,108,163 x 12] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ride\_id : num [1:1108163] 22178529 22178530 22178531 22178532 22178533 ...  
## $ started\_at : POSIXct[1:1108163], format: "2019-04-01 00:02:22" "2019-04-01 00:03:02" ...  
## $ ended\_at : POSIXct[1:1108163], format: "2019-04-01 00:09:48" "2019-04-01 00:20:30" ...  
## $ rideable\_type : num [1:1108163] 6251 6226 5649 4151 3270 ...  
## $ 01 - Rental Details Duration In Seconds Uncapped: num [1:1108163] 446 1048 252 357 1007 ...  
## $ start\_station\_id : num [1:1108163] 81 317 283 26 202 420 503 260 211 211 ...  
## $ start\_station\_name : chr [1:1108163] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle St & Jackson Blvd" "McClurg Ct & Illinois St" ...  
## $ end\_station\_id : num [1:1108163] 56 59 174 133 129 426 500 499 211 211 ...  
## $ end\_station\_name : chr [1:1108163] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...  
## $ member\_casual : chr [1:1108163] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...  
## $ Member Gender : chr [1:1108163] "Male" "Female" "Male" "Male" ...  
## $ 05 - Member Details Member Birthday Year : num [1:1108163] 1975 1984 1990 1993 1992 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. `01 - Rental Details Rental ID` = col\_double(),  
## .. `01 - Rental Details Local Start Time` = col\_datetime(format = ""),  
## .. `01 - Rental Details Local End Time` = col\_datetime(format = ""),  
## .. `01 - Rental Details Bike ID` = col\_double(),  
## .. `01 - Rental Details Duration In Seconds Uncapped` = col\_number(),  
## .. `03 - Rental Start Station ID` = col\_double(),  
## .. `03 - Rental Start Station Name` = col\_character(),  
## .. `02 - Rental End Station ID` = col\_double(),  
## .. `02 - Rental End Station Name` = col\_character(),  
## .. `User Type` = col\_character(),  
## .. `Member Gender` = col\_character(),  
## .. `05 - Member Details Member Birthday Year` = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

#### Convert ride\_id and rideable\_type to character using mutate from the dyplr library so that they can stack correctly in the new dataframe

q4\_2019 <- mutate(q4\_2019, ride\_id = as.character(ride\_id)  
 ,rideable\_type = as.character(rideable\_type))   
q3\_2019 <- mutate(q3\_2019, ride\_id = as.character(ride\_id)  
 ,rideable\_type = as.character(rideable\_type))   
q2\_2019 <- mutate(q2\_2019, ride\_id = as.character(ride\_id)  
 ,rideable\_type = as.character(rideable\_type))

#### Stack individual quarter’s data frames into one big data frame using bind\_rows from tidyverse package and dyplr library

all\_trips <- bind\_rows(q2\_2019, q3\_2019, q4\_2019, q1\_2020)

#### Check structure of new table

colnames(all\_trips)

## [1] "ride\_id"   
## [2] "started\_at"   
## [3] "ended\_at"   
## [4] "rideable\_type"   
## [5] "01 - Rental Details Duration In Seconds Uncapped"  
## [6] "start\_station\_id"   
## [7] "start\_station\_name"   
## [8] "end\_station\_id"   
## [9] "end\_station\_name"   
## [10] "member\_casual"   
## [11] "Member Gender"   
## [12] "05 - Member Details Member Birthday Year"   
## [13] "tripduration"   
## [14] "gender"   
## [15] "birthyear"   
## [16] "start\_lat"   
## [17] "start\_lng"   
## [18] "end\_lat"   
## [19] "end\_lng"

str(all\_trips)

## spec\_tbl\_df [3,879,822 x 19] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ride\_id : chr [1:3879822] "22178529" "22178530" "22178531" "22178532" ...  
## $ started\_at : POSIXct[1:3879822], format: "2019-04-01 00:02:22" "2019-04-01 00:03:02" ...  
## $ ended\_at : POSIXct[1:3879822], format: "2019-04-01 00:09:48" "2019-04-01 00:20:30" ...  
## $ rideable\_type : chr [1:3879822] "6251" "6226" "5649" "4151" ...  
## $ 01 - Rental Details Duration In Seconds Uncapped: num [1:3879822] 446 1048 252 357 1007 ...  
## $ start\_station\_id : num [1:3879822] 81 317 283 26 202 420 503 260 211 211 ...  
## $ start\_station\_name : chr [1:3879822] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle St & Jackson Blvd" "McClurg Ct & Illinois St" ...  
## $ end\_station\_id : num [1:3879822] 56 59 174 133 129 426 500 499 211 211 ...  
## $ end\_station\_name : chr [1:3879822] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...  
## $ member\_casual : chr [1:3879822] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...  
## $ Member Gender : chr [1:3879822] "Male" "Female" "Male" "Male" ...  
## $ 05 - Member Details Member Birthday Year : num [1:3879822] 1975 1984 1990 1993 1992 ...  
## $ tripduration : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ gender : chr [1:3879822] NA NA NA NA ...  
## $ birthyear : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ start\_lat : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ start\_lng : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ end\_lat : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ end\_lng : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. `01 - Rental Details Rental ID` = col\_double(),  
## .. `01 - Rental Details Local Start Time` = col\_datetime(format = ""),  
## .. `01 - Rental Details Local End Time` = col\_datetime(format = ""),  
## .. `01 - Rental Details Bike ID` = col\_double(),  
## .. `01 - Rental Details Duration In Seconds Uncapped` = col\_number(),  
## .. `03 - Rental Start Station ID` = col\_double(),  
## .. `03 - Rental Start Station Name` = col\_character(),  
## .. `02 - Rental End Station ID` = col\_double(),  
## .. `02 - Rental End Station Name` = col\_character(),  
## .. `User Type` = col\_character(),  
## .. `Member Gender` = col\_character(),  
## .. `05 - Member Details Member Birthday Year` = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

tibble(all\_trips)

## # A tibble: 3,879,822 x 19  
## ride\_id started\_at ended\_at rideable\_type  
## <chr> <dttm> <dttm> <chr>   
## 1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48 6251   
## 2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30 6226   
## 3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19 5649   
## 4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58 4151   
## 5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13 3270   
## 6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56 3123   
## 7 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41 6418   
## 8 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11 4513   
## 9 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44 3280   
## 10 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39 5534   
## # ... with 3,879,812 more rows, and 15 more variables:  
## # 01 - Rental Details Duration In Seconds Uncapped <dbl>,  
## # start\_station\_id <dbl>, start\_station\_name <chr>, end\_station\_id <dbl>,  
## # end\_station\_name <chr>, member\_casual <chr>, Member Gender <chr>,  
## # 05 - Member Details Member Birthday Year <dbl>, tripduration <dbl>,  
## # gender <chr>, birthyear <dbl>, start\_lat <dbl>, start\_lng <dbl>,  
## # end\_lat <dbl>, end\_lng <dbl>

#### Remove birthyear, tripduration, and gender fields as this data was dropped beginning in 2020

all\_trips <- all\_trips %>%   
 select(-c( birthyear, gender, "01 - Rental Details Duration In Seconds Uncapped", "05 - Member Details Member Birthday Year", "Member Gender", "tripduration"))

#### Inspect the new table that has been created

clean\_names(all\_trips) # Clean column names post rename for characters, numbers, and underscores only with clean\_names from the \*janitor\* package

## # A tibble: 3,879,822 x 13  
## ride\_id started\_at ended\_at rideable\_type  
## <chr> <dttm> <dttm> <chr>   
## 1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48 6251   
## 2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30 6226   
## 3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19 5649   
## 4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58 4151   
## 5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13 3270   
## 6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56 3123   
## 7 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41 6418   
## 8 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11 4513   
## 9 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44 3280   
## 10 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39 5534   
## # ... with 3,879,812 more rows, and 9 more variables: start\_station\_id <dbl>,  
## # start\_station\_name <chr>, end\_station\_id <dbl>, end\_station\_name <chr>,  
## # member\_casual <chr>, start\_lat <dbl>, start\_lng <dbl>, end\_lat <dbl>,  
## # end\_lng <dbl>

colnames(all\_trips) # List of column names

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

nrow(all\_trips) # How many rows are in data frame?

## [1] 3879822

dim(all\_trips) # Dimensions of the data frame?

## [1] 3879822 13

head(all\_trips) # See the first 6 rows of data frame

## # A tibble: 6 x 13  
## ride\_id started\_at ended\_at rideable\_type start\_station\_id  
## <chr> <dttm> <dttm> <chr> <dbl>  
## 1 221785~ 2019-04-01 00:02:22 2019-04-01 00:09:48 6251 81  
## 2 221785~ 2019-04-01 00:03:02 2019-04-01 00:20:30 6226 317  
## 3 221785~ 2019-04-01 00:11:07 2019-04-01 00:15:19 5649 283  
## 4 221785~ 2019-04-01 00:13:01 2019-04-01 00:18:58 4151 26  
## 5 221785~ 2019-04-01 00:19:26 2019-04-01 00:36:13 3270 202  
## 6 221785~ 2019-04-01 00:19:39 2019-04-01 00:23:56 3123 420  
## # ... with 8 more variables: start\_station\_name <chr>, end\_station\_id <dbl>,  
## # end\_station\_name <chr>, member\_casual <chr>, start\_lat <dbl>,  
## # start\_lng <dbl>, end\_lat <dbl>, end\_lng <dbl>

str(all\_trips) # See list of columns and data types (numeric, character, etc)

## tibble [3,879,822 x 13] (S3: tbl\_df/tbl/data.frame)  
## $ ride\_id : chr [1:3879822] "22178529" "22178530" "22178531" "22178532" ...  
## $ started\_at : POSIXct[1:3879822], format: "2019-04-01 00:02:22" "2019-04-01 00:03:02" ...  
## $ ended\_at : POSIXct[1:3879822], format: "2019-04-01 00:09:48" "2019-04-01 00:20:30" ...  
## $ rideable\_type : chr [1:3879822] "6251" "6226" "5649" "4151" ...  
## $ start\_station\_id : num [1:3879822] 81 317 283 26 202 420 503 260 211 211 ...  
## $ start\_station\_name: chr [1:3879822] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle St & Jackson Blvd" "McClurg Ct & Illinois St" ...  
## $ end\_station\_id : num [1:3879822] 56 59 174 133 129 426 500 499 211 211 ...  
## $ end\_station\_name : chr [1:3879822] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...  
## $ member\_casual : chr [1:3879822] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...  
## $ start\_lat : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ start\_lng : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ end\_lat : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ end\_lng : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...

summary(all\_trips) # Statistical summary of data. Mainly for numeric values

## ride\_id started\_at ended\_at   
## Length:3879822 Min. :2019-04-01 00:02:22 Min. :2019-04-01 00:09:48   
## Class :character 1st Qu.:2019-06-23 07:49:09 1st Qu.:2019-06-23 08:20:27   
## Mode :character Median :2019-08-14 17:43:38 Median :2019-08-14 18:02:04   
## Mean :2019-08-26 00:49:59 Mean :2019-08-26 01:14:37   
## 3rd Qu.:2019-10-12 12:10:21 3rd Qu.:2019-10-12 12:36:16   
## Max. :2020-03-31 23:51:34 Max. :2020-05-19 20:10:34   
##   
## rideable\_type start\_station\_id start\_station\_name end\_station\_id   
## Length:3879822 Min. : 1.0 Length:3879822 Min. : 1.0   
## Class :character 1st Qu.: 77.0 Class :character 1st Qu.: 77.0   
## Mode :character Median :174.0 Mode :character Median :174.0   
## Mean :202.9 Mean :203.8   
## 3rd Qu.:291.0 3rd Qu.:291.0   
## Max. :675.0 Max. :675.0   
## NA's :1   
## end\_station\_name member\_casual start\_lat start\_lng   
## Length:3879822 Length:3879822 Min. :42 Min. :-88   
## Class :character Class :character 1st Qu.:42 1st Qu.:-88   
## Mode :character Mode :character Median :42 Median :-88   
## Mean :42 Mean :-88   
## 3rd Qu.:42 3rd Qu.:-88   
## Max. :42 Max. :-88   
## NA's :3452935 NA's :3452935   
## end\_lat end\_lng   
## Min. :42 Min. :-88   
## 1st Qu.:42 1st Qu.:-88   
## Median :42 Median :-88   
## Mean :42 Mean :-88   
## 3rd Qu.:42 3rd Qu.:-88   
## Max. :42 Max. :-88   
## NA's :3452936 NA's :3452936

#### Issues to address regarding the new dataframe:

In the “member\_casual” column, there are two names for members (“member” and “Subscriber”) and two names for casual riders (“Customer” and “casual”). I will consolidate the data from four to two labels and use the same structure as q1\_2020, by replacing “Subscriber” with “member”, and “Customer” with “casual”. In order to complete the business task effectively and compare members and casual riders, I need to add some additional columns of data (day, month, year, hour) that I can derive from “started\_at” and “ended\_at”, to provide additional opportunities to aggregate the data for analysis. I will add a column for length of ride since the q1\_2020 data did not have the “tripduration” column. For consistency, I will add “ride\_length” to the entire dataframe, using “started\_at” and “ended\_at”. There are some rides where tripduration shows up as negative, including several hundred rides where I am making the assumption that Divvy took bikes out of circulation for Quality Assurance reasons. I will delete these rides in our new file.

#### Preview how many observations fall under each usertype

table(all\_trips$member\_casual)

##   
## casual Customer member Subscriber   
## 48480 857474 378407 2595461

#### Reassign to the desired values (using the q1\_2020 labels)

all\_trips <- all\_trips %>%   
 mutate(member\_casual = recode(member\_casual  
 ,"Subscriber" = "member"  
 ,"Customer" = "casual"))

#### Verify that the proper number of observations were reassigned

table(all\_trips$member\_casual)

##   
## casual member   
## 905954 2973868

#### Add columns that list the date, month, day, year, and hour of each ride

#### This will allow us to aggregate ride data for each month, day, or year

all\_trips$date <- as.Date(all\_trips$started\_at) #The default format is yyyy-mm-dd  
tibble(all\_trips$date)

## # A tibble: 3,879,822 x 1  
## `all\_trips$date`  
## <date>   
## 1 2019-04-01   
## 2 2019-04-01   
## 3 2019-04-01   
## 4 2019-04-01   
## 5 2019-04-01   
## 6 2019-04-01   
## 7 2019-04-01   
## 8 2019-04-01   
## 9 2019-04-01   
## 10 2019-04-01   
## # ... with 3,879,812 more rows

all\_trips$month <- format(as.Date(all\_trips$date), "%m")  
head(all\_trips$month)

## [1] "04" "04" "04" "04" "04" "04"

all\_trips$day <- format(as.Date(all\_trips$date), "%d")  
glimpse(all\_trips$day)

## chr [1:3879822] "01" "01" "01" "01" "01" "01" "01" "01" "01" "01" "01" ...

all\_trips$year <- format(as.Date(all\_trips$date), "%Y")  
head(all\_trips$year)

## [1] "2019" "2019" "2019" "2019" "2019" "2019"

all\_trips$day\_of\_week <- format(as.Date(all\_trips$date), "%A")  
glimpse(all\_trips$day\_of\_week)

## chr [1:3879822] "Monday" "Monday" "Monday" "Monday" "Monday" "Monday" ...

all\_trips$hour\_of\_day <- hour(all\_trips$started\_at)  
head(all\_trips$hour\_of\_day)

## [1] 0 0 0 0 0 0

#### Add a “ride\_length” calculation to all\_trips (in seconds)

all\_trips$ride\_length <- difftime(all\_trips$ended\_at,all\_trips$started\_at)

#### Inspect the structure of the columns

str(all\_trips)

## tibble [3,879,822 x 20] (S3: tbl\_df/tbl/data.frame)  
## $ ride\_id : chr [1:3879822] "22178529" "22178530" "22178531" "22178532" ...  
## $ started\_at : POSIXct[1:3879822], format: "2019-04-01 00:02:22" "2019-04-01 00:03:02" ...  
## $ ended\_at : POSIXct[1:3879822], format: "2019-04-01 00:09:48" "2019-04-01 00:20:30" ...  
## $ rideable\_type : chr [1:3879822] "6251" "6226" "5649" "4151" ...  
## $ start\_station\_id : num [1:3879822] 81 317 283 26 202 420 503 260 211 211 ...  
## $ start\_station\_name: chr [1:3879822] "Daley Center Plaza" "Wood St & Taylor St" "LaSalle St & Jackson Blvd" "McClurg Ct & Illinois St" ...  
## $ end\_station\_id : num [1:3879822] 56 59 174 133 129 426 500 499 211 211 ...  
## $ end\_station\_name : chr [1:3879822] "Desplaines St & Kinzie St" "Wabash Ave & Roosevelt Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...  
## $ member\_casual : chr [1:3879822] "member" "member" "member" "member" ...  
## $ start\_lat : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ start\_lng : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ end\_lat : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ end\_lng : num [1:3879822] NA NA NA NA NA NA NA NA NA NA ...  
## $ date : Date[1:3879822], format: "2019-04-01" "2019-04-01" ...  
## $ month : chr [1:3879822] "04" "04" "04" "04" ...  
## $ day : chr [1:3879822] "01" "01" "01" "01" ...  
## $ year : chr [1:3879822] "2019" "2019" "2019" "2019" ...  
## $ day\_of\_week : chr [1:3879822] "Monday" "Monday" "Monday" "Monday" ...  
## $ hour\_of\_day : int [1:3879822] 0 0 0 0 0 0 0 0 0 0 ...  
## $ ride\_length : 'difftime' num [1:3879822] 446 1048 252 357 ...  
## ..- attr(\*, "units")= chr "secs"

#### Convert “ride\_length” from Factor to numeric to 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

#### Remove “incomplete/bad” data

#### The dataframe includes a few hundred entries when bikes were taken out of docks and checked for quality by Divvy or ride\_length was negative

#### Create a new version of the dataframe (v2) since data is being removed/dropped

all\_trips\_v2 <- all\_trips[!(all\_trips$start\_station\_name == "HQ QR" | all\_trips$ride\_length<0),]

## Phase 4: Analyze Data to Answer Questions

#### Summary Statistics/Descriptive analysis on ride\_length (all figures in seconds)

mean(all\_trips\_v2$ride\_length) #average (total ride length / rides)

## [1] 1479.139

median(all\_trips\_v2$ride\_length) #midpoint number in the ascending array of ride lengths

## [1] 712

max(all\_trips\_v2$ride\_length) #longest ride

## [1] 9387024

min(all\_trips\_v2$ride\_length) #shortest ride

## [1] 1

#### Alternatively, use summary() on the specific attribute

summary(all\_trips\_v2$ride\_length)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1 412 712 1479 1289 9387024

#### Compare members and casual users

all\_trips\_v2 %>% group\_by(member\_casual) %>% summarize(mean\_ride\_length = mean(ride\_length))

## # A tibble: 2 x 2  
## member\_casual mean\_ride\_length  
## <chr> <dbl>  
## 1 casual 3553.  
## 2 member 850.

all\_trips\_v2 %>% group\_by(member\_casual) %>% summarize(mode\_ride\_length = mode(ride\_length))

## # A tibble: 2 x 2  
## member\_casual mode\_ride\_length  
## <chr> <chr>   
## 1 casual numeric   
## 2 member numeric

all\_trips\_v2 %>% group\_by(member\_casual) %>% summarize(max\_ride\_length = max(ride\_length))

## # A tibble: 2 x 2  
## member\_casual max\_ride\_length  
## <chr> <dbl>  
## 1 casual 9387024  
## 2 member 9056634

all\_trips\_v2 %>% group\_by(member\_casual) %>% summarize(min\_ride\_length = min(ride\_length))

## # A tibble: 2 x 2  
## member\_casual min\_ride\_length  
## <chr> <dbl>  
## 1 casual 2  
## 2 member 1

OR

all\_trips\_v2 %>% group\_by(member\_casual) %>% summarize(mean\_ride\_length = mean(ride\_length)  
 ,mode\_ride\_length = mode(ride\_length)  
 ,max\_ride\_length = max(ride\_length)  
 ,min\_ride\_length = min(ride\_length))

## # A tibble: 2 x 5  
## member\_casual mean\_ride\_length mode\_ride\_length max\_ride\_length  
## <chr> <dbl> <chr> <dbl>  
## 1 casual 3553. numeric 9387024  
## 2 member 850. numeric 9056634  
## # ... with 1 more variable: min\_ride\_length <dbl>

#### Observe the average ride time by each day for members vs casual users

all\_trips\_v2 %>% group\_by(member\_casual, day\_of\_week) %>% summarize(mean\_ride\_length = mean(ride\_length))

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

## # A tibble: 14 x 3  
## # Groups: member\_casual [2]  
## member\_casual day\_of\_week mean\_ride\_length  
## <chr> <chr> <dbl>  
## 1 casual Friday 3774.  
## 2 casual Monday 3372.  
## 3 casual Saturday 3332.  
## 4 casual Sunday 3581.  
## 5 casual Thursday 3683.  
## 6 casual Tuesday 3596.  
## 7 casual Wednesday 3719.  
## 8 member Friday 825.  
## 9 member Monday 843.  
## 10 member Saturday 969.  
## 11 member Sunday 920.  
## 12 member Thursday 824.  
## 13 member Tuesday 826.  
## 14 member Wednesday 824.

#### Order the days of the week

all\_trips\_v2$day\_of\_week <- ordered(all\_trips\_v2$day\_of\_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))

#### Observe the average ride time by each day for members vs casual users with ordered week

all\_trips\_v2 %>% group\_by(member\_casual, day\_of\_week) %>% summarize(mean\_ride\_length = mean(ride\_length))

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

## # A tibble: 14 x 3  
## # Groups: member\_casual [2]  
## member\_casual day\_of\_week mean\_ride\_length  
## <chr> <ord> <dbl>  
## 1 casual Sunday 3581.  
## 2 casual Monday 3372.  
## 3 casual Tuesday 3596.  
## 4 casual Wednesday 3719.  
## 5 casual Thursday 3683.  
## 6 casual Friday 3774.  
## 7 casual Saturday 3332.  
## 8 member Sunday 920.  
## 9 member Monday 843.  
## 10 member Tuesday 826.  
## 11 member Wednesday 824.  
## 12 member Thursday 824.  
## 13 member Friday 825.  
## 14 member Saturday 969.

#### analyze ridership data by type and weekday

all\_trips\_v2 %>%   
 group\_by(member\_casual, day\_of\_week) %>% #groups by usertype and weekday  
 summarize(number\_of\_rides = n() #calculates the number of rides and average duration   
 ,avg\_ride\_length = 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 x 4  
## # Groups: member\_casual [2]  
## member\_casual day\_of\_week number\_of\_rides avg\_ride\_length  
## <chr> <ord> <int> <dbl>  
## 1 casual Sunday 181293 3581.  
## 2 casual Monday 103296 3372.  
## 3 casual Tuesday 90510 3596.  
## 4 casual Wednesday 92457 3719.  
## 5 casual Thursday 102679 3683.  
## 6 casual Friday 122404 3774.  
## 7 casual Saturday 209543 3332.  
## 8 member Sunday 267965 920.  
## 9 member Monday 472196 843.  
## 10 member Tuesday 508445 826.  
## 11 member Wednesday 500329 824.  
## 12 member Thursday 484177 824.  
## 13 member Friday 452790 825.  
## 14 member Saturday 287958 969.

#### analyze riders by time of day

all\_trips\_v2 %>%   
 group\_by(member\_casual, hour\_of\_day) %>%   
 summarize(number\_of\_rides = n()  
 ,avg\_ride\_length = mean(ride\_length)) %>%   
 arrange(member\_casual, hour\_of\_day)

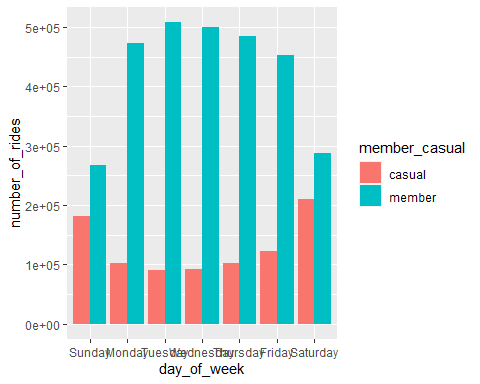
## `summarise()` has grouped output by 'member\_casual'. You can override using the `.groups` argument.

## # A tibble: 48 x 4  
## # Groups: member\_casual [2]  
## member\_casual hour\_of\_day number\_of\_rides avg\_ride\_length  
## <chr> <int> <int> <dbl>  
## 1 casual 0 8363 6256.  
## 2 casual 1 5495 6229.  
## 3 casual 2 3361 6232.  
## 4 casual 3 1982 10213.  
## 5 casual 4 1196 7592.  
## 6 casual 5 2690 5941.  
## 7 casual 6 6291 3984.  
## 8 casual 7 13302 1932.  
## 9 casual 8 22304 3289.  
## 10 casual 9 29057 4092.  
## # ... with 38 more rows

#### Viz for the number of rides by rider type - bar

all\_trips\_v2 %>%   
 group\_by(member\_casual, day\_of\_week) %>%   
 summarize(number\_of\_rides = n()  
 ,avg\_ride\_length = 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")

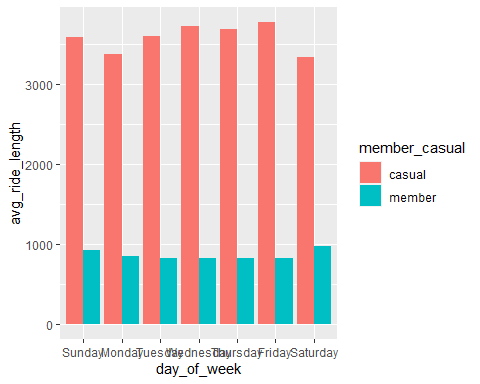
## `summarise()` has grouped output by 'member\_casual'. You can override using the `.groups` argument.



#### Viz for the avg ride length - bar

all\_trips\_v2 %>%   
 group\_by(member\_casual, day\_of\_week) %>%   
 summarize(number\_of\_rides = n()  
 ,avg\_ride\_length = mean(ride\_length)) %>%   
 arrange(member\_casual, day\_of\_week) %>%   
 ggplot(aes(x = day\_of\_week, y = avg\_ride\_length, fill = member\_casual)) +  
 geom\_col(position = "dodge")

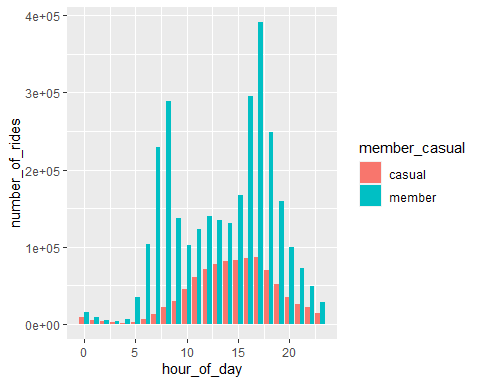
## `summarise()` has grouped output by 'member\_casual'. You can override using the `.groups` argument.



#### Viz for number of riders by rider type and hour of day - bar

all\_trips\_v2 %>%   
 group\_by(member\_casual, hour\_of\_day) %>%   
 summarize(number\_of\_rides = n()  
 ,avg\_ride\_length = mean(ride\_length)) %>%   
 arrange(member\_casual, hour\_of\_day) %>%   
 ggplot(aes(x = hour\_of\_day, y = number\_of\_rides, fill = member\_casual)) +   
 geom\_col(position = "dodge")

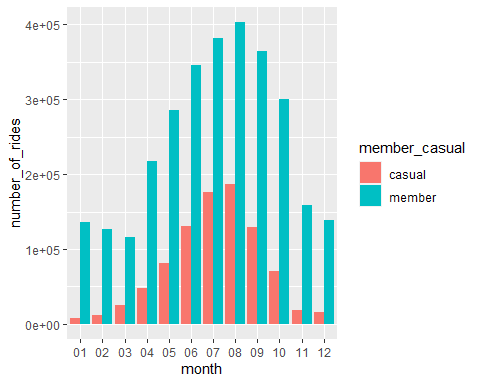
## `summarise()` has grouped output by 'member\_casual'. You can override using the `.groups` argument.



#### Viz by seasonality - bar

all\_trips\_v2 %>%   
 group\_by(member\_casual, month) %>%   
 summarize(number\_of\_rides = n()  
 ,avg\_ride\_length = mean(ride\_length)) %>%   
 arrange(member\_casual, month) %>%   
 ggplot(aes(x = month, 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.



## EXPORT CSV FILE FOR FURTHER ANALYSIS/SHARE PHASE IN TABLEAU

Create a csv file