Google Data Analytics Capstone Project_Cyclistic Bike-Share Analysis

Rodney Boyd

2022-06-04

Introduction

Bike-sharing systems are some of the world's rapidly growing -in number and popularity - urban innovations ((Wang, Lindsey, Schoner, & Harrison, 2016). This case study addresses the desire for the future growth of the fictional company, Cyclistic, a bike-share company launched in 2016 and based in Chicago. And, like most bike-sharing programs, the geo-tracked bicycles can be unlocked from one docking station and returned to any other docking station within their system of use at any time. This analyst has been tasked as a junior data analyst working with Cyclistic's in-house marketing analyst team currently working on the company's future success.

Cyclistic's bike-sharing program touts more than 5,800 bicycles and 600 docking stations. Included in those figures are the company's unique offerings of reclining bikes, hand tricycles, and cargo bikes purchased for the use of people with disabilities and those riders who cannot utilize traditional two-wheeled bicycles. Currently, 8% of Cyclistic's bike-share users opt for non-traditional two-wheeled bicycles. Furthermore, concerning bike utilization, most bike-share users utilize bicycles for leisure, and a minority (30%) use bicycles in their daily commute to work.

Up to the present time, Cyclistic's marketing strategy focused on developing general awareness and appealing to wide-ranging consumer segments by offering flexible pricing plans such as single-ride passes, full-day passes, and annual memberships. For this study, customers who purchase single-ride or full-day passes are casual riders, and customers who buy yearly memberships are Cyclistic members (Google, 2022).

Armed with the fact that annual members are much more profitable than casual riders, the marketing goal is to convert casual members to annual memberships. This junior data analyst has been tasked with gaining deep insight into how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics via an analysis of Cyclistic's historical bike trip data to identify trends (Google, 2022).

For the rest of this study, we will follow the formatting recommended phases from Google – Ask, Prepare, Process, Analyze, Share, and Act.

Ask

Google suggests that a problem is impossible to solve if you don't know what it is; furthermore, there are a few things to consider, such as defining the problem or identifying the business tasks, aligning stakeholder's expectations, remaining focused on the actual situation, maintaining communication with stakeholders, and staying mindful of context

(Google, 2022). In addition, Cyclistic posed the following three questions that they believe 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?

As a junior data analyst, I have been tasked with answering the following question:

• How do annual members and casual riders use Cyclistic bikes differently?

And I have been directed to produce the following deliverables:

1) A clear statement of the business tasks

2) A description of all data sources used

3) Documentation of any cleaning or manipulation of data

4) A summary of my analysis

5) Supporting visualization and key findings

6) My top three recommendations based on my analysis

The Business Task

The overarching business task is to identify the differences in how annual members and casual riders use Cyclistic bikes. By identifying these differences, Cyclistic will be able to develop a marketing strategy that will focus on converting casual riders into annual members to increase the number of annual members, which in turn will increase annual revenue.

The Key Stakeholders (reprinted, Google, 2022)

Lily Moreno: The director of marketing and your manager. Moreno is responsible for the development of campaigns and initiatives to promote the bike-share program. These may include email, social media, and other channels.

Cyclistic marketing analytics team: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy. You joined this team six months ago and have been busy learning about Cyclistic’s mission and business goals and how you, as a junior data analyst, can help Cyclistic achieve them.

Cyclistic executive team: The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

Prepare

In the Prepare phase, we must decide what data should be collected to answer the business questions and how to organize it so that it is functional. In addition, you must decide on which tools you will use, metrics to measure, the location of the data and database, and data security (Google, 2022).

As a junior data analyst, I have been tasked with the following duties:

1) Download data and store it appropriately.

2) Identify how it is organized.

3) Sort and filter the data.

4) Determine the credibility of the data.

R was used for the Prepare phase.

The data (the previous 12 months) was provided from Cyclistic's database at the following link:

https://divvy-tripdata.s3.amazonaws.com/index.html (https://divvy-tripdata.s3.amazonaws.com/index.html)

Importing the data into R

```
# Load library

library(data.table)

# Get a List of all files in directory named with a key word, say all `.csv` files

filenames <- list.files("C:/Users/rodne/Desktop/Cyclistic_Bike_Data/12_Month_Data", pattern="*.cs
v", full.names=TRUE)

# read and row bind all data sets

data <- rbindlist(lapply(filenames,fread))
```

```
View(data)
```

Now it is time to load a few packages before we clean and tidy the data and create a workable data table.

Now it is time to load a few packages before we clean and tidy the data and create a workable data table.

```
setwd("~")
library(dplyr)
library(lubridate)
library(readxl)
library(tidyr)
library(openxlsx)
library(tidyverse)
```

```
str(data)
```

```
## Classes 'data.table' and 'data.frame':   5667986 obs. of  13 variables:
##  $ ride_id           : chr  "CFA86D4455AA1030" "30D9DC61227D1AF3" "846D87A15682A284" "994D05AA
75A168F2" ...
##  $ rideable_type     : chr  "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct, format: "2021-03-16 08:32:30" "2021-03-28 01:26:28" ...
##  $ ended_at          : POSIXct, format: "2021-03-16 08:36:34" "2021-03-28 01:36:55" ...
##  $ start_station_name: chr  "Humboldt Blvd & Armitage Ave" "Humboldt Blvd & Armitage Ave" "Shi
elds Ave & 28th Pl" "Winthrop Ave & Lawrence Ave" ...
##  $ start_station_id  : chr  "15651" "15651" "15443" "TA1308000021" ...
##  $ end_station_name  : chr  "Stave St & Armitage Ave" "Central Park Ave & Bloomingdale Ave" "H
alsted St & 35th St" "Broadway & Sheridan Rd" ...
##  $ end_station_id    : chr  "13266" "18017" "TA1308000043" "13323" ...
##  $ start_lat         : num  41.9 41.9 41.8 42 42 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.8 42 42.1 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

I perfer dataframe summaries before I began my data cleaning. Hence, I forgot a package. But, first let us set the CRAN mirror/repository. I have had problems with running summarytools and missing Rtools in the past (rddrr.io, 2022).

```
get_compatible_rtools_version <- function(r_version = getRversion()) {
  require2("pkgbuild", min_version = "1.1.0")
  rtools_version_string <- pkgbuild::rtools_needed(r_version)
  if (!stringr::str_detect(rtools_version_string, "[\\d.]+")) {
    stop(
      "Can't determine compatible Rtools version, please report this at",
      "\nhttps://github.com/talgalili/installr/issues"
    )
  }
  str_extract(rtools_version_string, "[\\d.]+")
}

get_rtools_url <- function(rtools_version = get_compatible_rtools_version(),
                           arch = R.version$arch) {
  base_url <- "https://cran.r-project.org/bin/windows/Rtools/"
  rtools_version_nodots <- str_replace(rtools_version, "\\.", "")
  if (rtools_version < 4.0) {
    filename <- str_glue("Rtools{rtools_version_nodots}.exe")
  } else {
    filename <- str_glue("rtools{rtools_version_nodots}-{arch}.exe")
  }
  url <- str_glue("{base_url}{filename}")
  url
}



#' @title Downloads and installs Rtools
#' @aliases install.rtools
#' @description Install compatible version of Rtools for Windows.
#' By default, the function searches if a compatible Rtools is installed,
#' if not, it offers to install the latest compatible version.
#' @details
#' Rtools is a collection of software for building packages for R under Microsoft Windows,
#' or for building R itself (version 1.9.0 or later).
#' The original collection was put together by Prof. Brian Ripley;
#' it is currently being maintained by Duncan Murdoch.
#' @param check checks if we need to install Rtools or not.
#' @param check_r_update checks if there is an R update available (ignores patch versions),
#' if so, asks if user wants to install the R update first. (defaults to TRUE)
#' @param GUI Should a GUI be used when asking the user questions? (defaults to TRUE)
#' @param ... extra parameters to pass to \link{install.URL}
#' @return invisible(TRUE/FALSE) - was the installation successful or not.
#' @export
#' @references Rtools homepage: \url{https://cran.r-project.org/bin/windows/Rtools/}
#' @examples
#' \dontrun{
#' # installs the latest compatible version of Rtools if a compatible version is not yet installe
d
#' install.Rtools()
#' # (re)installs the latest compatible version of Rtools
#' install.Rtools(check = F)
#'
```

```
#' # skip R version check
#' install.Rtools(checkRupdate = F)
#' }
install.Rtools <- function(check = TRUE, check_r_update = TRUE, GUI = TRUE, ...) {
  if (check_r_update && r_update_available(ignore_patchlevel = T)) {
    update_r <- ask.user.yn.question(
      "A newer R version is available, do you want to update R first?", GUI = GUI)
    if (update_r) {
      updateR(GUI = GUI, print_R_versions = F, install_R = T, start_new_R = T, quit_R = F)
      print("Please run installr::install.Rtools using the newly installed R")
      return(invisible(FALSE))
    }
  }

  # Use pkgbuild to check if matching rtools is installed
  if (check) {
    require2("pkgbuild")
    found_rtools <- pkgbuild::has_rtools()
    if (found_rtools) {
      cat("No need to install Rtools - You've got the relevant version of Rtools installed\n")
      return(invisible(FALSE))
    }
  }

  # if we reached here - it means we'll need to install Rtools.
  tryCatch({
    rtools_url <- get_rtools_url()
    install.URL(rtools_url, ...)
  },
  error = function(e) {
    message("You'll need to go to the site and download this yourself.",
            " I'm now going to try and open the url for you.")
    browseURL("https://cran.r-project.org/bin/windows/Rtools/")
    stop(e)
  }
  )
}
```

```
local({r <- getOption("repos")
       r["CRAN"] <- "http://cran.r-project.org"
       options(repos=r)})
```

```
install.packages("summarytools",repos = "http://cran.us.r-project.org")
```

```
## package 'summarytools' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\rodne\AppData\Local\Temp\RtmpmAbi35\downloaded_packages
```

```
library(summarytools)
```

Summary Tools is good for frequencies tables, cross-tabulations, descriptive statistics, and dataframe summaries. In *Recommendation for Using summarytools with Rmarkdown* (Comtois, 2020), it has been recommended to specify ASCII to false to print good results when working with summarytools.

```
plain.ascii = FALSE
```

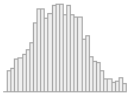
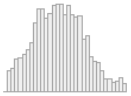
```
print(dfSummary(data, graph.magnif = 0.75), method = 'render')
```

Data Frame Summary

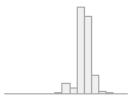
data

Dimensions: 5667986 x 13
Duplicates: 0

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	ride_id [character]	1. 00000123F60251E6 2. 000002EBE159AE82 3. 0000080D43BAA9E4 4. 00000B4F1F71F9C2 5. 00000CAE95438C9D 6. 00000EBBC119168C 7. 000019B7F053D461 8. 00001A81D056B01B 9. 00001B4F79D102B5 10. 00001BEE76AB24E0 [5667976 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 5667976 (100.0%)		5667986 (100.0%)	0 (0.0%)
2	rideable_type [character]	1. classic_bike 2. docked_bike 3. electric_bike	3268797 (57.7%) 311288 (5.5%) 2087901 (36.8%)		5667986 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
3	started_at [POSIXct, POSIXt]	min : 2021-03-01 00:01:09 med : 2021-08-07 19:12:50 max : 2022-02-28 23:58:44 range : 11m 27d 23H 57M 35S	4747127 distinct values		5667986 (100.0%)	0 (0.0%)
4	ended_at [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-07 19:35:11 max : 2022-03-01 08:55:17 range : 1y 0m 0d 8H 48M 49S	4740417 distinct values		5667986 (100.0%)	0 (0.0%)
5	start_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Michigan Ave & Oak St 4. Wells St & Concord Ln 5. Millennium Park 6. Clark St & Elm St 7. Wells St & Elm St 8. Theater on the Lake 9. Kingsbury St & Kinzie St 10. Clark St & Lincoln Ave [844 others]	712978 (12.6%) 82954 (1.5%) 44409 (0.8%) 43969 (0.8%) 42399 (0.7%) 41316 (0.7%) 37895 (0.7%) 36768 (0.6%) 34616 (0.6%) 33346 (0.6%) 4557336 (80.4%)		5667986 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
6	start_station_id [character]	1. (Empty string) 2. 13022 3. LF-005 4. 13300 5. 13042 6. TA1308000050 7. 13008 8. TA1307000039 9. KA1504000135 10. TA1308000001 [835 others]	712975 (12.6%) 82954 (1.5%) 47856 (0.8%) 46176 (0.8%) 44409 (0.8%) 43969 (0.8%) 42399 (0.7%) 41316 (0.7%) 37895 (0.7%) 36768 (0.6%) 4531269 (79.9%)		5667986 (100.0%)	0 (0.0%)
7	end_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Michigan Ave & Oak St 4. Wells St & Concord Ln 5. Millennium Park 6. Clark St & Elm St 7. Wells St & Elm St 8. Theater on the Lake 9. Kingsbury St & Kinzie St 10. Wabash Ave & Grand Ave [845 others]	761817 (13.4%) 83648 (1.5%) 44913 (0.8%) 44149 (0.8%) 43082 (0.8%) 40599 (0.7%) 37550 (0.7%) 36952 (0.7%) 33848 (0.6%) 33504 (0.6%) 4507924 (79.5%)		5667986 (100.0%)	0 (0.0%)
8	end_station_id [character]	1. (Empty string) 2. 13022 3. LF-005 4. 13042 5. 13300 6. TA1308000050 7. 13008 8. TA1307000039 9. KA1504000135 10. TA1308000001 [837 others]	761817 (13.4%) 83648 (1.5%) 53932 (1.0%) 44913 (0.8%) 44325 (0.8%) 44149 (0.8%) 43082 (0.8%) 40599 (0.7%) 37550 (0.7%) 36952 (0.7%) 4477019 (79.0%)		5667986 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
9	start_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 45.6 IQR (CV) : 0 (0)	413530 distinct values		5667986 (100.0%)	0 (0.0%)
10	start_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -73.8 IQR (CV) : 0 (0)	392879 distinct values		5667986 (100.0%)	0 (0.0%)
11	end_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.4 ≤ 41.9 ≤ 42.2 IQR (CV) : 0 (0)	440966 distinct values		5663369 (99.9%)	4617 (0.1%)
12	end_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -89 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	401665 distinct values		5663369 (99.9%)	4617 (0.1%)
13	member_casual [character]	1. casual 2. member	2540693 (44.8%) 3127293 (55.2%)		5667986 (100.0%)	0 (0.0%)

Generated by summarytools (https://github.com/dcomtois/summarytools) 1.0.1 (R (https://www.r-project.org/) version 4.1.2)
2022-06-21

The above data frame summary gives us an overview of the dataset before data manipulation. We also could have used other functions such as dim,and colnames.

```
dim(data)
```

```
## [1] 5667986      13
```

```
colnames(data)
```

```
##  [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"
```

As shown above, a dataframe summary is efficient too.

Process

In the Process phase, we must ensure data integrity – verifying that the data is accurate and consistent. In addition, we must choose the tool or tools we will be working with, ensure that the data is clean and ready to analyze, and document the results.

As a junior data analyst, I have been tasked with the following duties:

Check the data for errors.

Choose your tools.

Transform the data so you can work with it effectively

Document the cleaning process and manipulation of data.

R was used for the Process phase. This phase requires fixing data entry errors, missing data, finding and removing NA' and Nulls, and removing duplicates.

Removing duplicates is done here; in addtion to, another row and column count.

```
dim(data)
```

```
## [1] 5667986      13
```

Let us begin this phase by quickly removing the duplicates.

```
data %>% distinct()
```

```
##                  ride_id rideable_type          started_at            ended_at
##       1: CFA86D4455AA1030  classic_bike 2021-03-16 08:32:30 2021-03-16 08:36:34
##       2: 30D9DC61227D1AF3  classic_bike 2021-03-28 01:26:28 2021-03-28 01:36:55
##       3: 846D87A15682A284  classic_bike 2021-03-11 21:17:29 2021-03-11 21:33:53
##       4: 994D05AA75A168F2  classic_bike 2021-03-11 13:26:42 2021-03-11 13:55:41
##       5: DF7464FBE92D8308  classic_bike 2021-03-21 09:09:37 2021-03-21 09:27:33
##      ---
## 5667982: 211BE0DC162D85B7 electric_bike 2022-02-23 17:47:49 2022-02-23 18:02:29
## 5667983: D4D53E78000C8CA1 electric_bike 2022-02-04 10:43:47 2022-02-04 10:50:52
## 5667984: 9E85F07D2F94492B electric_bike 2022-02-28 09:16:33 2022-02-28 09:28:11
## 5667985: B61B559F81F1D823 electric_bike 2022-02-10 16:55:16 2022-02-10 16:57:53
## 5667986: 841C701610CF0609 electric_bike 2022-02-21 16:35:20 2022-02-21 16:42:53
##                    start_station_name start_station_id
##       1: Humboldt Blvd & Armitage Ave            15651
##       2: Humboldt Blvd & Armitage Ave            15651
##       3:        Shields Ave & 28th Pl            15443
##       4:  Winthrop Ave & Lawrence Ave     TA1308000021
##       5:     Glenwood Ave & Touhy Ave              525
##      ---
## 5667982:
## 5667983:
## 5667984:        Wood St & Chicago Ave              637
## 5667985:
## 5667986:
##                            end_station_name end_station_id start_lat start_lng
##       1:             Stave St & Armitage Ave          13266  41.91751 -87.70181
##       2: Central Park Ave & Bloomingdale Ave          18017  41.91751 -87.70181
##       3:                Halsted St & 35th St   TA1308000043  41.84273 -87.63549
##       4:              Broadway & Sheridan Rd          13323  41.96881 -87.65766
##       5:           Chicago Ave & Sheridan Rd           E008  42.01270 -87.66606
##      ---
## 5667982:            Leavitt St & Chicago Ave          18058  41.88000 -87.63000
## 5667983:            Leavitt St & Chicago Ave          18058  41.91000 -87.68000
## 5667984:                 Canal St & Adams St          13011  41.89571 -87.67221
## 5667985:                 Canal St & Adams St          13011  41.88000 -87.63000
## 5667986:                Larrabee St & Oak St   KA1504000116  41.88000 -87.65000
##           end_lat   end_lng member_casual
##       1: 41.91774 -87.69139        casual
##       2: 41.91417 -87.71676        casual
##       3: 41.83066 -87.64717        casual
##       4: 41.95283 -87.64999        casual
##       5: 42.05049 -87.67782        casual
##      ---
## 5667982: 41.89550 -87.68202        member
## 5667983: 41.89550 -87.68202        member
## 5667984: 41.87926 -87.63990        member
## 5667985: 41.87926 -87.63990        member
## 5667986: 41.90022 -87.64299        member
```

```
# Check the number of observations
dim(data)
```

```
## [1] 5667986      13
```

The same amount of observations. Now its time to do a little work.

```
str(data)
```

```
## Classes 'data.table' and 'data.frame':   5667986 obs. of  13 variables:
##  $ ride_id           : chr  "CFA86D4455AA1030" "30D9DC61227D1AF3" "846D87A15682A284" "994D05AA
75A168F2" ...
##  $ rideable_type     : chr  "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct, format: "2021-03-16 08:32:30" "2021-03-28 01:26:28" ...
##  $ ended_at          : POSIXct, format: "2021-03-16 08:36:34" "2021-03-28 01:36:55" ...
##  $ start_station_name: chr  "Humboldt Blvd & Armitage Ave" "Humboldt Blvd & Armitage Ave" "Shi
elds Ave & 28th Pl" "Winthrop Ave & Lawrence Ave" ...
##  $ start_station_id  : chr  "15651" "15651" "15443" "TA1308000021" ...
##  $ end_station_name  : chr  "Stave St & Armitage Ave" "Central Park Ave & Bloomingdale Ave" "H
alsted St & 35th St" "Broadway & Sheridan Rd" ...
##  $ end_station_id    : chr  "13266" "18017" "TA1308000043" "13323" ...
##  $ start_lat         : num  41.9 41.9 41.8 42 42 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.8 42 42.1 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Here we are creating date variables, or checking on them.

```
data <- data %>% mutate(start_date = ymd_hms(started_at))
data <- data %>% mutate(end_date = ymd_hms(ended_at))

data <- data %>%
  mutate(
    Start_Yr = year(started_at),
    Start_Mth = month(started_at),
    Start_Day = wday(started_at),
    Start_Hr = hour(started_at)
  )

data <- data %>%
  mutate(
    End_Yr = year(ended_at),
    End_Mth = month(ended_at),
    End_Day = wday(ended_at),
    End_Hr = hour(ended_at)
  )
```

Creating a variable of how long a person held the bike. In addition, changing the variable into minutes instead of seconds. And, checking for negative time values.

```
data %>%
  filter(ended_at < started_at) %>%
  count()
```

```
##      n
## 1: 145
```

```
data <- data %>% filter(ended_at > started_at)
```

```
dim(data)
```

```
## [1] 5667333      23
```

```
data <- data %>%
  mutate(held_time = ended_at - started_at)

data <- data %>%
  mutate(held_time2 = as.numeric(ended_at - started_at, units="mins"))
```

Let us look at the columns

```
str(data)
```

```
## Classes 'data.table' and 'data.frame':   5667333 obs. of  25 variables:
##  $ ride_id           : chr  "CFA86D4455AA1030" "30D9DC61227D1AF3" "846D87A15682A284" "994D05AA
75A168F2" ...
##  $ rideable_type     : chr  "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct, format: "2021-03-16 08:32:30" "2021-03-28 01:26:28" ...
##  $ ended_at          : POSIXct, format: "2021-03-16 08:36:34" "2021-03-28 01:36:55" ...
##  $ start_station_name: chr  "Humboldt Blvd & Armitage Ave" "Humboldt Blvd & Armitage Ave" "Shi
elds Ave & 28th Pl" "Winthrop Ave & Lawrence Ave" ...
##  $ start_station_id  : chr  "15651" "15651" "15443" "TA1308000021" ...
##  $ end_station_name  : chr  "Stave St & Armitage Ave" "Central Park Ave & Bloomingdale Ave" "H
alsted St & 35th St" "Broadway & Sheridan Rd" ...
##  $ end_station_id    : chr  "13266" "18017" "TA1308000043" "13323" ...
##  $ start_lat         : num  41.9 41.9 41.8 42 42 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.8 42 42.1 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
##  $ start_date        : POSIXct, format: "2021-03-16 08:32:30" "2021-03-28 01:26:28" ...
##  $ end_date          : POSIXct, format: "2021-03-16 08:36:34" "2021-03-28 01:36:55" ...
##  $ Start_Yr          : num  2021 2021 2021 2021 2021 ...
##  $ Start_Mth         : num  3 3 3 3 3 3 3 3 3 ...
##  $ Start_Day         : num  3 1 5 5 1 7 7 3 4 5 ...
##  $ Start_Hr          : int  8 1 21 13 9 11 14 7 15 17 ...
##  $ End_Yr            : num  2021 2021 2021 2021 2021 ...
##  $ End_Mth           : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ End_Day           : num  3 1 5 5 1 7 7 3 4 5 ...
##  $ End_Hr            : int  8 1 21 13 9 11 14 8 15 17 ...
##  $ held_time         : 'difftime' num  244 627 984 1739 ...
##   ..- attr(*, "units")= chr "secs"
##  $ held_time2        : num  4.07 10.45 16.4 28.98 17.93 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
head(data)
```

```
##             ride_id rideable_type          started_at            ended_at
## 1: CFA86D4455AA1030  classic_bike 2021-03-16 08:32:30 2021-03-16 08:36:34
## 2: 30D9DC61227D1AF3  classic_bike 2021-03-28 01:26:28 2021-03-28 01:36:55
## 3: 846D87A15682A284  classic_bike 2021-03-11 21:17:29 2021-03-11 21:33:53
## 4: 994D05AA75A168F2  classic_bike 2021-03-11 13:26:42 2021-03-11 13:55:41
## 5: DF7464FBE92D8308  classic_bike 2021-03-21 09:09:37 2021-03-21 09:27:33
## 6: CEBA8516FD17F8D8  classic_bike 2021-03-20 11:08:47 2021-03-20 11:29:39
##              start_station_name start_station_id
## 1: Humboldt Blvd & Armitage Ave            15651
## 2: Humboldt Blvd & Armitage Ave            15651
## 3:        Shields Ave & 28th Pl            15443
## 4:  Winthrop Ave & Lawrence Ave     TA1308000021
## 5:     Glenwood Ave & Touhy Ave              525
## 6:     Glenwood Ave & Touhy Ave              525
##                        end_station_name end_station_id start_lat start_lng
## 1:             Stave St & Armitage Ave          13266  41.91751 -87.70181
## 2: Central Park Ave & Bloomingdale Ave          18017  41.91751 -87.70181
## 3:                Halsted St & 35th St   TA1308000043  41.84273 -87.63549
## 4:              Broadway & Sheridan Rd          13323  41.96881 -87.65766
## 5:           Chicago Ave & Sheridan Rd           E008  42.01270 -87.66606
## 6:           Chicago Ave & Sheridan Rd           E008  42.01270 -87.66606
##     end_lat   end_lng member_casual          start_date            end_date
## 1: 41.91774 -87.69139        casual 2021-03-16 08:32:30 2021-03-16 08:36:34
## 2: 41.91417 -87.71676        casual 2021-03-28 01:26:28 2021-03-28 01:36:55
## 3: 41.83066 -87.64717        casual 2021-03-11 21:17:29 2021-03-11 21:33:53
## 4: 41.95283 -87.64999        casual 2021-03-11 13:26:42 2021-03-11 13:55:41
## 5: 42.05049 -87.67782        casual 2021-03-21 09:09:37 2021-03-21 09:27:33
## 6: 42.05049 -87.67782        casual 2021-03-20 11:08:47 2021-03-20 11:29:39
##    Start_Yr Start_Mth Start_Day Start_Hr End_Yr End_Mth End_Day End_Hr
## 1:     2021         3         3        8   2021       3       3      8
## 2:     2021         3         1        1   2021       3       1      1
## 3:     2021         3         5       21   2021       3       5     21
## 4:     2021         3         5       13   2021       3       5     13
## 5:     2021         3         1        9   2021       3       1      9
## 6:     2021         3         7       11   2021       3       7     11
##    held_time held_time2
## 1:  244 secs   4.066667
## 2:  627 secs  10.450000
## 3:  984 secs  16.400000
## 4: 1739 secs  28.983333
## 5: 1076 secs  17.933333
## 6: 1252 secs  20.866667
```

Taking a moment to address missing data

```
sum(is.na(data))
```

```
## [1] 9234
```

```
summary(is.na(data))
```

```
##   ride_id        rideable_type   started_at       ended_at
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:5667333   FALSE:5667333   FALSE:5667333   FALSE:5667333
##
##  start_station_name start_station_id end_station_name end_station_id
##  Mode :logical      Mode :logical    Mode :logical    Mode :logical
##  FALSE:5667333      FALSE:5667333    FALSE:5667333    FALSE:5667333
##
##  start_lat       start_lng        end_lat         end_lng
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:5667333   FALSE:5667333   FALSE:5662716   FALSE:5662716
##                                  TRUE :4617      TRUE :4617
##  member_casual   start_date       end_date        Start_Yr
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:5667333   FALSE:5667333   FALSE:5667333   FALSE:5667333
##
##  Start_Mth       Start_Day        Start_Hr         End_Yr
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:5667333   FALSE:5667333   FALSE:5667333   FALSE:5667333
##
##   End_Mth         End_Day          End_Hr        held_time
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:5667333   FALSE:5667333   FALSE:5667333   FALSE:5667333
##
##  held_time2
##  Mode :logical
##  FALSE:5667333
##
```

We have 4,617 NAs in end_lat and end_lng

```
colSums(is.na(data))
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##                  0                  0                  0                  0
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0               4617               4617
##      member_casual         start_date           end_date           Start_Yr
##                  0                  0                  0                  0
##          Start_Mth          Start_Day           Start_Hr            End_Yr
##                  0                  0                  0                  0
##            End_Mth            End_Day             End_Hr          held_time
##                  0                  0                  0                  0
##         held_time2
##                  0
```

```
data <- data %>% filter(is.na(end_lat)==F)
```

Looking to see if NA's are removed

```
colSums(is.na(data))
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   start_station_id   end_station_name     end_station_id
##                  0                  0                  0                  0
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0                  0                  0
##      member_casual         start_date           end_date           Start_Yr
##                  0                  0                  0                  0
##          Start_Mth          Start_Day           Start_Hr             End_Yr
##                  0                  0                  0                  0
##            End_Mth            End_Day             End_Hr          held_time
##                  0                  0                  0                  0
##         held_time2
##                  0
```

In the end_lat and end_lng variables, we had 4,617 cases of missing data. We just removed them.

Let us check on how far a person as traveled when the use the bicycle

Calculate distance latitude longitude

Using distVincentyEllipsoid = uses an ellipsoid in meters by default

Using distGeo: Distance on an ellipsoid (the geodesic) Highly accurate estimate of the shortest distance between two points on an ellipsoid

```
installed.packages("geosphere")
```

```
##      Package LibPath Version Priority Depends Imports LinkingTo Suggests
##      Enhances License License_is_FOSS License_restricts_use OS_type Archs
##      MD5sum NeedsCompilation Built
```

```
library(geosphere)
library(ggplot2)


data$distHaversine <- round(distHaversine(data[,c("start_lng", "start_lat")], data[,c("end_lng",
"end_lat")])/1000, 3)


data$distVincentyEllipsoid <- round(distVincentyEllipsoid(data[,c("start_lng", "start_lat")], dat
a[,c("end_lng", "end_lat")])/1000, 3)


data$distGeo <- round(distGeo(data[,c("start_lng", "start_lat")], data[,c("end_lng", "end_lat")],
a=6378137, f=1/298.257223563)/1000, 3)
```

```
head(data)
```

```
##             ride_id rideable_type          started_at            ended_at
## 1: CFA86D4455AA1030  classic_bike 2021-03-16 08:32:30 2021-03-16 08:36:34
## 2: 30D9DC61227D1AF3  classic_bike 2021-03-28 01:26:28 2021-03-28 01:36:55
## 3: 846D87A15682A284  classic_bike 2021-03-11 21:17:29 2021-03-11 21:33:53
## 4: 994D05AA75A168F2  classic_bike 2021-03-11 13:26:42 2021-03-11 13:55:41
## 5: DF7464FBE92D8308  classic_bike 2021-03-21 09:09:37 2021-03-21 09:27:33
## 6: CEBA8516FD17F8D8  classic_bike 2021-03-20 11:08:47 2021-03-20 11:29:39
##              start_station_name start_station_id
## 1: Humboldt Blvd & Armitage Ave            15651
## 2: Humboldt Blvd & Armitage Ave            15651
## 3:        Shields Ave & 28th Pl            15443
## 4:  Winthrop Ave & Lawrence Ave     TA1308000021
## 5:     Glenwood Ave & Touhy Ave              525
## 6:     Glenwood Ave & Touhy Ave              525
##                        end_station_name end_station_id start_lat start_lng
## 1:             Stave St & Armitage Ave          13266  41.91751 -87.70181
## 2: Central Park Ave & Bloomingdale Ave          18017  41.91751 -87.70181
## 3:                Halsted St & 35th St   TA1308000043  41.84273 -87.63549
## 4:              Broadway & Sheridan Rd          13323  41.96881 -87.65766
## 5:           Chicago Ave & Sheridan Rd           E008  42.01270 -87.66606
## 6:           Chicago Ave & Sheridan Rd           E008  42.01270 -87.66606
##     end_lat   end_lng member_casual          start_date            end_date
## 1: 41.91774 -87.69139        casual 2021-03-16 08:32:30 2021-03-16 08:36:34
## 2: 41.91417 -87.71676        casual 2021-03-28 01:26:28 2021-03-28 01:36:55
## 3: 41.83066 -87.64717        casual 2021-03-11 21:17:29 2021-03-11 21:33:53
## 4: 41.95283 -87.64999        casual 2021-03-11 13:26:42 2021-03-11 13:55:41
## 5: 42.05049 -87.67782        casual 2021-03-21 09:09:37 2021-03-21 09:27:33
## 6: 42.05049 -87.67782        casual 2021-03-20 11:08:47 2021-03-20 11:29:39
##    Start_Yr Start_Mth Start_Day Start_Hr End_Yr End_Mth End_Day End_Hr
## 1:     2021         3         3        8   2021       3       3      8
## 2:     2021         3         1        1   2021       3       1      1
## 3:     2021         3         5       21   2021       3       5     21
## 4:     2021         3         5       13   2021       3       5     13
## 5:     2021         3         1        9   2021       3       1      9
## 6:     2021         3         7       11   2021       3       7     11
##    held_time held_time2 distHaversine distVincentyEllipsoid distGeo
## 1:  244 secs   4.066667         0.863                0.865   0.865
## 2:  627 secs  10.450000         1.293                1.294   1.294
## 3:  984 secs  16.400000         1.657                1.655   1.655
## 4: 1739 secs  28.983333         1.889                1.885   1.885
## 5: 1076 secs  17.933333         4.318                4.309   4.309
## 6: 1252 secs  20.866667         4.318                4.309   4.309
```

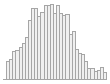
```
plain.ascii = FALSE
```

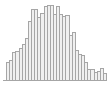
```
print(dfSummary(data, graph.magnif = 0.75), method = 'render')
```

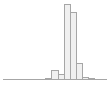
Data Frame Summary

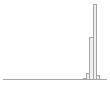
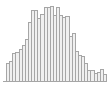
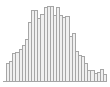
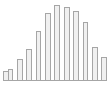
data

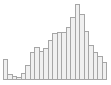
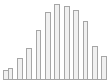
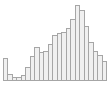
Dimensions: 5662716 x 28
Duplicates: 0

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	ride_id [character]	1. 00000123F60251E6 2. 000002EBE159AE82 3. 0000080D43BAA9E4 4. 00000B4F1F71F9C2 5. 00000CAE95438C9D 6. 00000EBBC119168C 7. 000019B7F053D461 8. 00001A81D056B01B 9. 00001B4F79D102B5 10. 00001BEE76AB24E0 [5662706 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 5662706 (100.0%)		5662716 (100.0%)	0 (0.0%)
2	rideable_type [character]	1. classic_bike 2. docked_bike 3. electric_bike	3264229 (57.6%) 310951 (5.5%) 2087536 (36.9%)		5662716 (100.0%)	0 (0.0%)
3	started_at [POSIXct, POSIXt]	min : 2021-03-01 00:01:09 med : 2021-08-07 19:15:11 max : 2022-02-28 23:58:44 range : 11m 27d 23H 57M 35S	4743338 distinct values		5662716 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
4	ended_at [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-07 19:37:27 max : 2022-03-01 08:55:17 range : 1y 0m 0d 8H 48M 49S	4736551 distinct values		5662716 (100.0%)	0 (0.0%)
5	start_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Michigan Ave & Oak St 4. Wells St & Concord Ln 5. Millennium Park 6. Clark St & Elm St 7. Wells St & Elm St 8. Theater on the Lake 9. Kingsbury St & Kinzie St 10. Clark St & Lincoln Ave [844 others]	712953 (12.6%) 82847 (1.5%) 44368 (0.8%) 43940 (0.8%) 42299 (0.7%) 41280 (0.7%) 37865 (0.7%) 36730 (0.6%) 34595 (0.6%) 33326 (0.6%) 4552513 (80.4%)		5662716 (100.0%)	0 (0.0%)
6	start_station_id [character]	1. (Empty string) 2. 13022 3. LF-005 4. 13300 5. 13042 6. TA1308000050 7. 13008 8. TA1307000039 9. KA1504000135 10. TA1308000001 [835 others]	712950 (12.6%) 82847 (1.5%) 47819 (0.8%) 46116 (0.8%) 44368 (0.8%) 43940 (0.8%) 42299 (0.7%) 41280 (0.7%) 37865 (0.7%) 36730 (0.6%) 4526502 (79.9%)		5662716 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
7	end_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Michigan Ave & Oak St 4. Wells St & Concord Ln 5. Millennium Park 6. Clark St & Elm St 7. Wells St & Elm St 8. Theater on the Lake 9. Kingsbury St & Kinzie St 10. Wabash Ave & Grand Ave [845 others]	756757 (13.4%) 83640 (1.5%) 44912 (0.8%) 44147 (0.8%) 43082 (0.8%) 40598 (0.7%) 37550 (0.7%) 36951 (0.7%) 33846 (0.6%) 33504 (0.6%) 4507729 (79.6%)		5662716 (100.0%)	0 (0.0%)
8	end_station_id [character]	1. (Empty string) 2. 13022 3. LF-005 4. 13042 5. 13300 6. TA1308000050 7. 13008 8. TA1307000039 9. KA1504000135 10. TA1308000001 [837 others]	756757 (13.4%) 83640 (1.5%) 53930 (1.0%) 44912 (0.8%) 44325 (0.8%) 44147 (0.8%) 43082 (0.8%) 40598 (0.7%) 37550 (0.7%) 36951 (0.7%) 4476824 (79.1%)		5662716 (100.0%)	0 (0.0%)
9	start_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 45.6 IQR (CV) : 0 (0)	413481 distinct values		5662716 (100.0%)	0 (0.0%)
10	start_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -73.8 IQR (CV) : 0 (0)	392830 distinct values		5662716 (100.0%)	0 (0.0%)
11	end_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.4 ≤ 41.9 ≤ 42.2 IQR (CV) : 0 (0)	440957 distinct values		5662716 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
12	end_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -89 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	401659 distinct values		5662716 (100.0%)	0 (0.0%)
13	member_casual [character]	1. casual 2. member	2536826 (44.8%) 3125890 (55.2%)		5662716 (100.0%)	0 (0.0%)
14	start_date [POSIXct, POSIXt]	min : 2021-03-01 00:01:09 med : 2021-08-07 19:15:11 max : 2022-02-28 23:58:44 range : 11m 27d 23H 57M 35S	4743338 distinct values		5662716 (100.0%)	0 (0.0%)
15	end_date [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-07 19:37:27 max : 2022-03-01 08:55:17 range : 1y 0m 0d 8H 48M 49S	4736551 distinct values		5662716 (100.0%)	0 (0.0%)
16	Start_Yr [numeric]	Min : 2021 Mean : 2021 Max : 2022	2021 : 5443510 (96.1%) 2022 : 219206 (3.9%)		5662716 (100.0%)	0 (0.0%)
17	Start_Mth [numeric]	Mean (sd) : 7.3 (2.6) min ≤ med ≤ max: 1 ≤ 7 ≤ 12 IQR (CV) : 3 (0.3)	12 distinct values		5662716 (100.0%)	0 (0.0%)
18	Start_Day [numeric]	Mean (sd) : 4.1 (2.1) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 866493 (15.3%) 2 : 722763 (12.8%) 3 : 755358 (13.3%) 4 : 766937 (13.5%) 5 : 745653 (13.2%) 6 : 814644 (14.4%) 7 : 990868 (17.5%)		5662716 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
19	Start_Hr [integer]	Mean (sd) : 14.2 (5.1) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 7 (0.4)	24 distinct values		5662716 (100.0%)	0 (0.0%)
20	End_Yr [numeric]	Min : 2021 Mean : 2021 Max : 2022	2021 : 5443439 (96.1%) 2022 : 219277 (3.9%)		5662716 (100.0%)	0 (0.0%)
21	End_Mth [numeric]	Mean (sd) : 7.3 (2.6) min ≤ med ≤ max: 1 ≤ 7 ≤ 12 IQR (CV) : 3 (0.3)	12 distinct values		5662716 (100.0%)	0 (0.0%)
22	End_Day [numeric]	Mean (sd) : 4.1 (2.1) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 872842 (15.4%) 2 : 724107 (12.8%) 3 : 755360 (13.3%) 4 : 767007 (13.5%) 5 : 744620 (13.1%) 6 : 810895 (14.3%) 7 : 987885 (17.4%)		5662716 (100.0%)	0 (0.0%)
23	End_Hr [integer]	Mean (sd) : 14.4 (5.1) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 7 (0.4)	24 distinct values		5662716 (100.0%)	0 (0.0%)
24	held_time [difftime]	min : 1 med : 711 max : 3356649 units : secs	24909 distinct values		5662716 (100.0%)	0 (0.0%)
25	held_time2 [numeric]	Mean (sd) : 20.8 (166.4) min ≤ med ≤ max: 0 ≤ 11.8 ≤ 55944.2 IQR (CV) : 14.9 (8)	24909 distinct values		5662716 (100.0%)	0 (0.0%)
26	distHaversine [numeric]	Mean (sd) : 2.2 (2) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 1190.9 IQR (CV) : 2 (0.9)	17098 distinct values		5662716 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
27	distVincentyEllipsoid [numeric]	Mean (sd) : 2.2 (2) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 1192.2 IQR (CV) : 2 (0.9)	17074 distinct values		5662716 (100.0%)	0 (0.0%)
28	distGeo [numeric]	Mean (sd) : 2.2 (2) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 1192.2 IQR (CV) : 2 (0.9)	17074 distinct values		5662716 (100.0%)	0 (0.0%)

Generated by summarytools (https://github.com/dcomtois/summarytools) 1.0.1 (R (https://www.r-project.org/) version 4.1.2)
2022-06-21

It would appear that we have a few outliers in the variable held_time2. It would appear someone held on to a bicycle for over 38 hours. Let us remove the outlier.

```
dim(data)
```

```
## [1] 5662716      28
```

```
install.packages("ggstatsplot")
```

```
## package 'ggstatsplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\rodne\AppData\Local\Temp\RtmpmAbi35\downloaded_packages
```

```
library(ggstatsplot)

Q <- quantile(data$held_time2, probs=c(.25, .75), na.rm = FALSE)
```

```
iqr <- IQR(data$held_time2)
```

```
up <-  Q[2]+1.5*iqr # Upper Range
low<- Q[1]-1.5*iqr # Lower Range
```

```
eliminated<- subset(data, data$held_time2 > (Q[1] - 1.5*iqr) & data$held_time2 < (Q[2]+1.5*iqr))
```

```
dim(data)
```

```
## [1] 5662716      28
```

```
dim(eliminated)
```

```
## [1] 5243900      28
```

Our dataset with the outliers removed is now called eliminated. Let us change that.

```
data2 <- eliminated
```

```
plain.ascii = FALSE
```

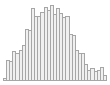
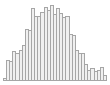
```
print(dfSummary(data2, graph.magnif = 0.75), method = 'render')
```

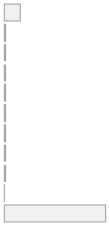
Data Frame Summary

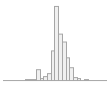
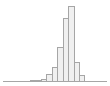
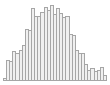
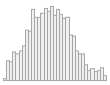
data2

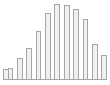
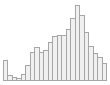
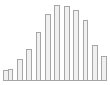
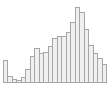
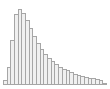
Dimensions: 5243900 x 28
Duplicates: 0

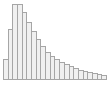
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	ride_id [character]	1. 00000123F60251E6 2. 000002EBE159AE82 3. 00000B4F1F71F9C2 4. 00000EBBC119168C 5. 000019B7F053D461 6. 00001A81D056B01B 7. 00001B4F79D102B5 8. 00001BEE76AB24E0 9. 00001DCF2BC423F4 10. 000020C92AA9D6F7 [5243890 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 5243890 (100.0%)		5243900 (100.0%)	0 (0.0%)
2	rideable_type [character]	1. classic_bike 2. docked_bike 3. electric_bike	3055014 (58.3%) 210316 (4.0%) 1978570 (37.7%)		5243900 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
3	started_at [POSIXct, POSIXt]	min : 2021-03-01 00:05:42 med : 2021-08-10 10:41:14 max : 2022-02-28 23:58:44 range : 11m 27d 23H 53M 2S	4459962 distinct values		5243900 (100.0%)	0 (0.0%)
4	ended_at [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-10 10:55:03 max : 2022-03-01 00:20:36 range : 1y 0m 0d 0H 14M 8S	4454349 distinct values		5243900 (100.0%)	0 (0.0%)
5	start_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Wells St & Concord Ln 4. Clark St & Elm St 5. Wells St & Elm St 6. Michigan Ave & Oak St 7. Kingsbury St & Kinzie St 8. Theater on the Lake 9. Clark St & Lincoln Ave 10. Millennium Park [844 others]	680112 (13.0%) 64057 (1.2%) 42342 (0.8%) 39726 (0.8%) 36762 (0.7%) 35261 (0.7%) 33899 (0.6%) 31825 (0.6%) 31189 (0.6%) 30608 (0.6%) 4218119 (80.4%)		5243900 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
6	start_station_id [character]	1. (Empty string) 2. 13022 3. LF-005 4. TA1308000050 5. TA1307000039 6. KA1504000135 7. 13042 8. 13300 9. KA1503000043 10. TA1308000001 [835 others]	680109 (13.0%) 64057 (1.2%) 42494 (0.8%) 42342 (0.8%) 39726 (0.8%) 36762 (0.7%) 35261 (0.7%) 34629 (0.7%) 33899 (0.6%) 31825 (0.6%) 4202796 (80.1%)		5243900 (100.0%)	0 (0.0%)
7	end_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Wells St & Concord Ln 4. Clark St & Elm St 5. Wells St & Elm St 6. Michigan Ave & Oak St 7. Kingsbury St & Kinzie St 8. Millennium Park 9. Clark St & Lincoln Ave 10. Theater on the Lake [845 others]	712919 (13.6%) 62347 (1.2%) 42664 (0.8%) 39030 (0.7%) 36637 (0.7%) 35014 (0.7%) 33271 (0.6%) 32030 (0.6%) 30813 (0.6%) 30724 (0.6%) 4188451 (79.9%)		5243900 (100.0%)	0 (0.0%)
8	end_station_id [character]	1. (Empty string) 2. 13022 3. LF-005 4. TA1308000050 5. TA1307000039 6. KA1504000135 7. 13042 8. KA1503000043 9. 13300 10. 13008 [837 others]	712919 (13.6%) 62347 (1.2%) 47322 (0.9%) 42664 (0.8%) 39030 (0.7%) 36637 (0.7%) 35014 (0.7%) 33271 (0.6%) 32520 (0.6%) 32030 (0.6%) 4170146 (79.5%)		5243900 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
9	start_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 45.6 IQR (CV) : 0 (0)	403350 distinct values		5243900 (100.0%)	0 (0.0%)
10	start_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -73.8 IQR (CV) : 0 (0)	383999 distinct values		5243900 (100.0%)	0 (0.0%)
11	end_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 42.1 IQR (CV) : 0 (0)	429697 distinct values		5243900 (100.0%)	0 (0.0%)
12	end_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.9 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	391622 distinct values		5243900 (100.0%)	0 (0.0%)
13	member_casual [character]	1. casual 2. member	2181367 (41.6%) 3062533 (58.4%)		5243900 (100.0%)	0 (0.0%)
14	start_date [POSIXct, POSIXt]	min : 2021-03-01 00:05:42 med : 2021-08-10 10:41:14 max : 2022-02-28 23:58:44 range : 11m 27d 23H 53M 2S	4459962 distinct values		5243900 (100.0%)	0 (0.0%)
15	end_date [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-10 10:55:03 max : 2022-03-01 00:20:36 range : 1y 0m 0d 0H 14M 8S	4454349 distinct values		5243900 (100.0%)	0 (0.0%)
16	Start_Yr [numeric]	Min : 2021 Mean : 2021 Max : 2022	2021 : 5029654 (95.9%) 2022 : 214246 (4.1%)		5243900 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
17	Start_Mth [numeric]	Mean (sd) : 7.4 (2.6) min ≤ med ≤ max: 1 ≤ 8 ≤ 12 IQR (CV) : 3 (0.3)	12 distinct values		5243900 (100.0%)	0 (0.0%)
18	Start_Day [numeric]	Mean (sd) : 4.1 (2) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 766490 (14.6%) 2 : 671780 (12.8%) 3 : 714864 (13.6%) 4 : 728904 (13.9%) 5 : 708877 (13.5%) 6 : 762875 (14.5%) 7 : 890110 (17.0%)		5243900 (100.0%)	0 (0.0%)
19	Start_Hr [integer]	Mean (sd) : 14.2 (5.1) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 7 (0.4)	24 distinct values		5243900 (100.0%)	0 (0.0%)
20	End_Yr [numeric]	Min : 2021 Mean : 2021 Max : 2022	2021 : 5029626 (95.9%) 2022 : 214274 (4.1%)		5243900 (100.0%)	0 (0.0%)
21	End_Mth [numeric]	Mean (sd) : 7.4 (2.6) min ≤ med ≤ max: 1 ≤ 8 ≤ 12 IQR (CV) : 3 (0.3)	12 distinct values		5243900 (100.0%)	0 (0.0%)
22	End_Day [numeric]	Mean (sd) : 4.1 (2.1) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 771006 (14.7%) 2 : 672306 (12.8%) 3 : 714700 (13.6%) 4 : 728661 (13.9%) 5 : 708097 (13.5%) 6 : 760520 (14.5%) 7 : 888610 (16.9%)		5243900 (100.0%)	0 (0.0%)
23	End_Hr [integer]	Mean (sd) : 14.3 (5.1) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 7 (0.4)	24 distinct values		5243900 (100.0%)	0 (0.0%)
24	held_time [difftime]	min : 1 med : 657 max : 2629 units : secs	2629 distinct values		5243900 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
25	held_time2 [numeric]	Mean (sd) : 13.6 (9.5) min ≤ med ≤ max: 0 ≤ 10.9 ≤ 43.8 IQR (CV) : 12.2 (0.7)	2629 distinct values		5243900 (100.0%)	0 (0.0%)
26	distHaversine [numeric]	Mean (sd) : 2.1 (1.8) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 1190.9 IQR (CV) : 1.9 (0.9)	13391 distinct values		5243900 (100.0%)	0 (0.0%)
27	distVincentyEllipsoid [numeric]	Mean (sd) : 2.1 (1.8) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 1192.2 IQR (CV) : 1.9 (0.9)	13375 distinct values		5243900 (100.0%)	0 (0.0%)
28	distGeo [numeric]	Mean (sd) : 2.1 (1.8) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 1192.2 IQR (CV) : 1.9 (0.9)	13375 distinct values		5243900 (100.0%)	0 (0.0%)

Generated by summarytools (https://github.com/dcomtois/summarytools) 1.0.1 (R (https://www.r-project.org/) version 4.1.2)
2022-06-21

It would also appear that someone traveled over 1,100 miles on the bicycle as well. This too would appear to be an outlier.

```
Q <- quantile(data2$distGeo, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(data2$distGeo)
up <-  Q[2]+1.5*iqr # Upper Range
low<- Q[1]-1.5*iqr # Lower Range

eliminated2<- subset(data2, data2$distGeo > (Q[1] - 1.5*iqr) & data2$distGeo < (Q[2]+1.5*iqr))
```

```
dim(eliminated2)
```

```
## [1] 4982661      28
```

```
data3 <- eliminated2
```

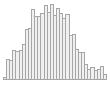
```
plain.ascii = FALSE
```

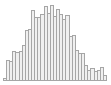
```
print(dfSummary(data3, graph.magnif = 0.75), method = 'render')
```

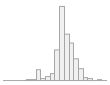
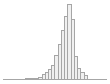
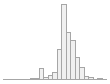
Data Frame Summary

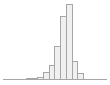
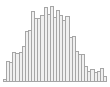
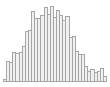
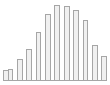
data3

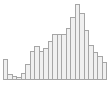
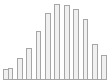
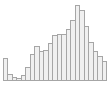
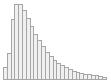
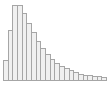
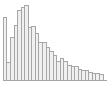
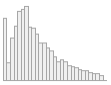
Dimensions: 4982661 x 28
Duplicates: 0

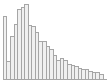
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	ride_id [character]	1. 00000123F60251E6 2. 000002EBE159AE82 3. 00000B4F1F71F9C2 4. 00000EBBC119168C 5. 000019B7F053D461 6. 00001A81D056B01B 7. 00001B4F79D102B5 8. 00001BEE76AB24E0 9. 00001DCF2BC423F4 10. 000020C92AA9D6F7 [4982651 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 4982651 (100.0%)		4982661 (100.0%)	0 (0.0%)
2	rideable_type [character]	1. classic_bike 2. docked_bike 3. electric_bike	2944937 (59.1%) 203734 (4.1%) 1833990 (36.8%)		4982661 (100.0%)	0 (0.0%)
3	started_at [POSIXct, POSIXt]	min : 2021-03-01 00:05:42 med : 2021-08-10 13:20:54 max : 2022-02-28 23:58:38 range : 11m 27d 23H 52M 56S	4271443 distinct values		4982661 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
4	ended_at [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-10 13:32:39 max : 2022-03-01 00:20:28 range : 1y 0m 0d 0H 14M 0S	4266002 distinct values		4982661 (100.0%)	0 (0.0%)
5	start_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Wells St & Concord Ln 4. Clark St & Elm St 5. Wells St & Elm St 6. Kingsbury St & Kinzie St 7. Michigan Ave & Oak St 8. Clark St & Lincoln Ave 9. Clark St & Armitage Ave 10. Wells St & Huron St [843 others]	624960 (12.5%) 59655 (1.2%) 41507 (0.8%) 38626 (0.8%) 36148 (0.7%) 33266 (0.7%) 31899 (0.6%) 30376 (0.6%) 29897 (0.6%) 29677 (0.6%) 4026650 (80.8%)		4982661 (100.0%)	0 (0.0%)
6	start_station_id [character]	1. (Empty string) 2. 13022 3. TA1308000050 4. LF-005 5. TA1307000039 6. KA1504000135 7. KA1503000043 8. 13300 9. 13042 10. 13179 [834 others]	624958 (12.5%) 59655 (1.2%) 41507 (0.8%) 40476 (0.8%) 38626 (0.8%) 36148 (0.7%) 33266 (0.7%) 32477 (0.7%) 31899 (0.6%) 30376 (0.6%) 4013273 (80.5%)		4982661 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
7	end_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Wells St & Concord Ln 4. Clark St & Elm St 5. Wells St & Elm St 6. Kingsbury St & Kinzie St 7. Michigan Ave & Oak St 8. Millennium Park 9. Clark St & Lincoln Ave 10. Dearborn St & Erie St [844 others]	664479 (13.3%) 58287 (1.2%) 41866 (0.8%) 38031 (0.8%) 36204 (0.7%) 32732 (0.7%) 31826 (0.6%) 30621 (0.6%) 30067 (0.6%) 29653 (0.6%) 3988895 (80.1%)		4982661 (100.0%)	0 (0.0%)
8	end_station_id [character]	1. (Empty string) 2. 13022 3. LF-005 4. TA1308000050 5. TA1307000039 6. KA1504000135 7. KA1503000043 8. 13042 9. 13008 10. 13179 [836 others]	664479 (13.3%) 58287 (1.2%) 45094 (0.9%) 41866 (0.8%) 38031 (0.8%) 36204 (0.7%) 32732 (0.7%) 31826 (0.6%) 30621 (0.6%) 30067 (0.6%) 3973454 (79.7%)		4982661 (100.0%)	0 (0.0%)
9	start_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 42.1 IQR (CV) : 0 (0)	390381 distinct values		4982661 (100.0%)	0 (0.0%)
10	start_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	371411 distinct values		4982661 (100.0%)	0 (0.0%)
11	end_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 42.1 IQR (CV) : 0 (0)	412613 distinct values		4982661 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
12	end_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	374985 distinct values		4982661 (100.0%)	0 (0.0%)
13	member_casual [character]	1. casual 2. member	2080663 (41.8%) 2901998 (58.2%)		4982661 (100.0%)	0 (0.0%)
14	start_date [POSIXct, POSIXt]	min : 2021-03-01 00:05:42 med : 2021-08-10 13:20:54 max : 2022-02-28 23:58:38 range : 11m 27d 23H 52M 56S	4271443 distinct values		4982661 (100.0%)	0 (0.0%)
15	end_date [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-10 13:32:39 max : 2022-03-01 00:20:28 range : 1y 0m 0d 0H 14M 0S	4266002 distinct values		4982661 (100.0%)	0 (0.0%)
16	Start_Yr [numeric]	Min : 2021 Mean : 2021 Max : 2022	2021 : 4776076 (95.9%) 2022 : 206585 (4.1%)		4982661 (100.0%)	0 (0.0%)
17	Start_Mth [numeric]	Mean (sd) : 7.4 (2.6) min ≤ med ≤ max: 1 ≤ 8 ≤ 12 IQR (CV) : 3 (0.4)	12 distinct values		4982661 (100.0%)	0 (0.0%)
18	Start_Day [numeric]	Mean (sd) : 4.1 (2) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 727227 (14.6%) 2 : 638526 (12.8%) 3 : 678882 (13.6%) 4 : 692000 (13.9%) 5 : 673612 (13.5%) 6 : 725686 (14.6%) 7 : 846728 (17.0%)		4982661 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
19	Start_Hr [integer]	Mean (sd) : 14.2 (5.1) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 7 (0.4)	24 distinct values		4982661 (100.0%)	0 (0.0%)
20	End_Yr [numeric]	Min : 2021 Mean : 2021 Max : 2022	2021 : 4776050 (95.9%) 2022 : 206611 (4.1%)		4982661 (100.0%)	0 (0.0%)
21	End_Mth [numeric]	Mean (sd) : 7.4 (2.6) min ≤ med ≤ max: 1 ≤ 8 ≤ 12 IQR (CV) : 3 (0.4)	12 distinct values		4982661 (100.0%)	0 (0.0%)
22	End_Day [numeric]	Mean (sd) : 4.1 (2.1) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 731361 (14.7%) 2 : 639020 (12.8%) 3 : 678722 (13.6%) 4 : 691771 (13.9%) 5 : 672908 (13.5%) 6 : 723535 (14.5%) 7 : 845344 (17.0%)		4982661 (100.0%)	0 (0.0%)
23	End_Hr [integer]	Mean (sd) : 14.3 (5.1) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 7 (0.4)	24 distinct values		4982661 (100.0%)	0 (0.0%)
24	held_time [difftime]	min : 1 med : 625 max : 2629 units : secs	2629 distinct values		4982661 (100.0%)	0 (0.0%)
25	held_time2 [numeric]	Mean (sd) : 12.7 (8.8) min ≤ med ≤ max: 0 ≤ 10.4 ≤ 43.8 IQR (CV) : 10.9 (0.7)	2629 distinct values		4982661 (100.0%)	0 (0.0%)
26	distHaversine [numeric]	Mean (sd) : 1.8 (1.3) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 5.6 IQR (CV) : 1.7 (0.7)	5626 distinct values		4982661 (100.0%)	0 (0.0%)
27	distVincentyEllipsoid [numeric]	Mean (sd) : 1.8 (1.3) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 5.6 IQR (CV) : 1.7 (0.7)	5613 distinct values		4982661 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
28	distGeo [numeric]	Mean (sd) : 1.8 (1.3) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 5.6 IQR (CV) : 1.7 (0.7)	5613 distinct values		4982661 (100.0%)	0 (0.0%)

Generated by summarytools (https://github.com/dcomtois/summarytools) 1.0.1 (R (https://www.r-project.org/) version 4.1.2)
2022-06-21

It would appear that we took care of the outliers.

```
dim(data3)
```

```
## [1] 4982661      28
```

Analyze

In the Analyze phase, we must identify trends and relationships and determine how the findings and insights will help answer your business questions.

As a junior data analyst, I have been tasked with the following duties:

Aggregate the data so that it is useful and accessible.

Organize and format your data.

Perform claculations.

Identify trends and relationships.

R was used for the Analyze phase. This phase requires a summary of our analysis.

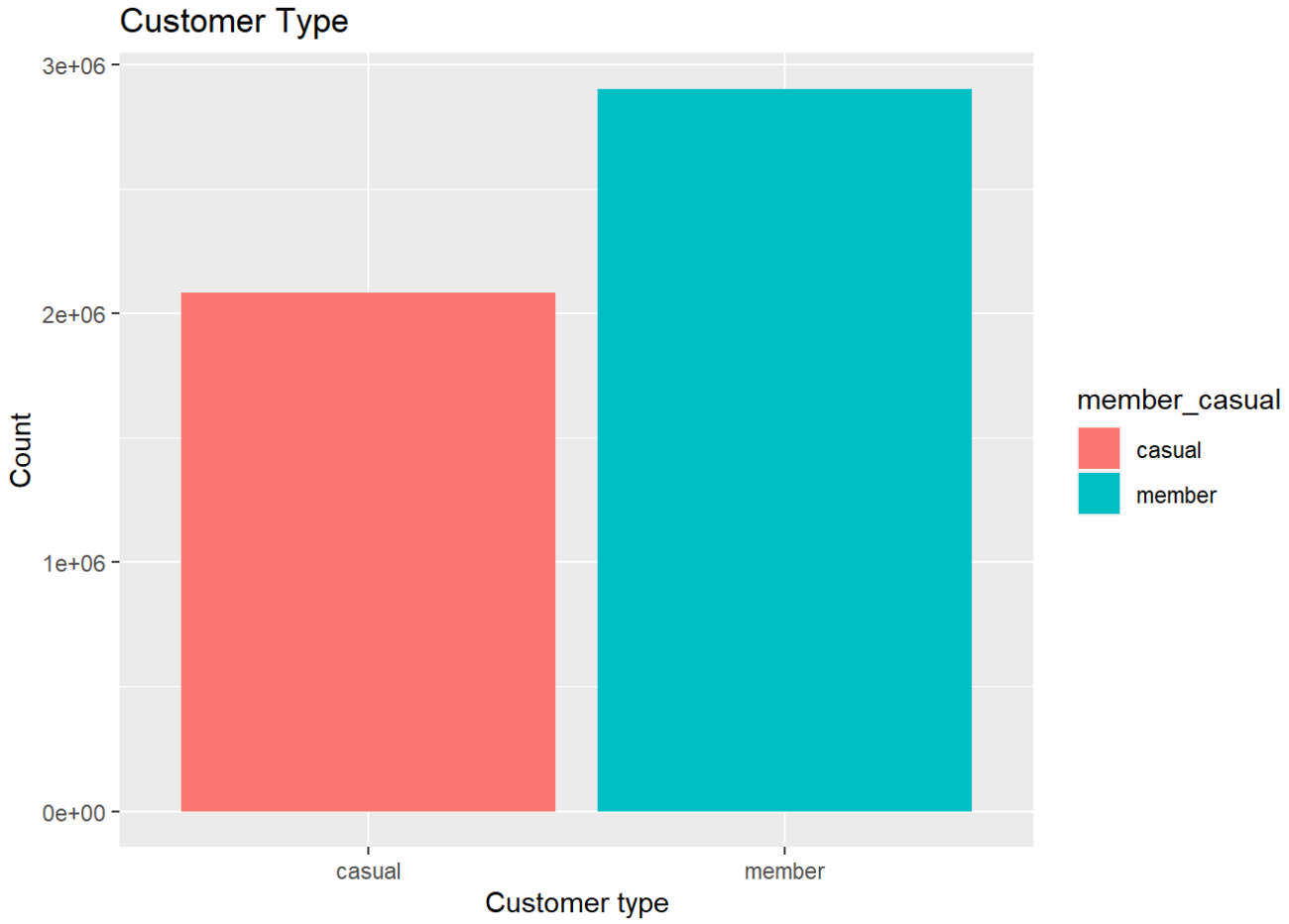
Let us begin by comparing members and casual users

```
df_members <- data3 %>%
  select(member_casual) %>%
  group_by(member_casual) %>%
  count() %>%
  arrange()
head(df_members)
```

```
## # A tibble: 2 x 2
## # Groups:   member_casual [2]
##   member_casual       n
##   <chr>           <int>
## 1 casual        2080663
## 2 member        2901998
```

Graphing the number of Member and Casual Users

```
ggplot(data = data3) + geom_bar(mapping = aes(x = member_casual, fill = member_casual)) + labs(ti
tle = "Customer Type", x = "Customer type", y = "Count")
```



Looking at the lenght of time a user utilize the bicycle.

```
aggregate(data3$held_time2  ~ data3$member_casual, FUN = mean)
```

```
##   data3$member_casual data3$held_time2
## 1              casual         15.25559
## 2              member         10.83797
```

```
aggregate(data3$held_time2  ~ data3$member_casual, FUN = median)
```

```
##   data3$member_casual data3$held_time2
## 1              casual        13.066667
## 2              member         8.816667
```

```
aggregate(data3$held_time2  ~ data3$member_casual, FUN = max)
```

```
##   data3$member_casual data3$held_time2
## 1              casual         43.81667
## 2              member         43.81667
```

```
aggregate(data3$held_time2  ~ data3$member_casual, FUN = min)
```

```
##   data3$member_casual data3$held_time2
## 1              casual       0.01666667
## 2              member       0.01666667
```

Let us add some names to the day of the week. And, create a column called day_of_week and month.

```
data4 <- data3
data4$day_of_week <- weekdays(data4$start_date)
```

```
data4 <- data4 %>%
  mutate(month = month(start_date, label = TRUE, abbr = TRUE))
```

```
str(data4)
```

```
## Classes 'data.table' and 'data.frame':   4982661 obs. of  30 variables:
##  $ ride_id             : chr  "CFA86D4455AA1030" "30D9DC61227D1AF3" "846D87A15682A284" "994D0
5AA75A168F2" ...
##  $ rideable_type       : chr  "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at          : POSIXct, format: "2021-03-16 08:32:30" "2021-03-28 01:26:28" ...
##  $ ended_at            : POSIXct, format: "2021-03-16 08:36:34" "2021-03-28 01:36:55" ...
##  $ start_station_name  : chr  "Humboldt Blvd & Armitage Ave" "Humboldt Blvd & Armitage Ave"
"Shields Ave & 28th Pl" "Winthrop Ave & Lawrence Ave" ...
##  $ start_station_id    : chr  "15651" "15651" "15443" "TA1308000021" ...
##  $ end_station_name    : chr  "Stave St & Armitage Ave" "Central Park Ave & Bloomingdale Ave"
"Halsted St & 35th St" "Broadway & Sheridan Rd" ...
##  $ end_station_id      : chr  "13266" "18017" "TA1308000043" "13323" ...
##  $ start_lat           : num  41.9 41.9 41.8 42 42 ...
##  $ start_lng           : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat             : num  41.9 41.9 41.8 42 42.1 ...
##  $ end_lng             : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual       : chr  "casual" "casual" "casual" "casual" ...
##  $ start_date          : POSIXct, format: "2021-03-16 08:32:30" "2021-03-28 01:26:28" ...
##  $ end_date            : POSIXct, format: "2021-03-16 08:36:34" "2021-03-28 01:36:55" ...
##  $ Start_Yr            : num  2021 2021 2021 2021 2021 ...
##  $ Start_Mth           : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ Start_Day           : num  3 1 5 5 1 7 7 3 4 5 ...
##  $ Start_Hr            : int  8 1 21 13 9 11 14 7 15 17 ...
##  $ End_Yr              : num  2021 2021 2021 2021 2021 ...
##  $ End_Mth             : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ End_Day             : num  3 1 5 5 1 7 7 3 4 5 ...
##  $ End_Hr              : int  8 1 21 13 9 11 14 8 15 17 ...
##  $ held_time           : 'difftime' num  244 627 984 1739 ...
##   ..- attr(*, "units")= chr "secs"
##  $ held_time2          : num  4.07 10.45 16.4 28.98 17.93 ...
##  $ distHaversine       : num  0.863 1.293 1.657 1.889 4.318 ...
##  $ distVincentyEllipsoid: num  0.865 1.294 1.655 1.885 4.309 ...
##  $ distGeo             : num  0.865 1.294 1.655 1.885 4.309 ...
##  $ day_of_week         : chr  "Tuesday" "Sunday" "Thursday" "Thursday" ...
##  $ month               : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<..: 3 3 3 3 3 3 3 3 3 3
...
##  - attr(*, ".internal.selfref")=<externalptr>
```

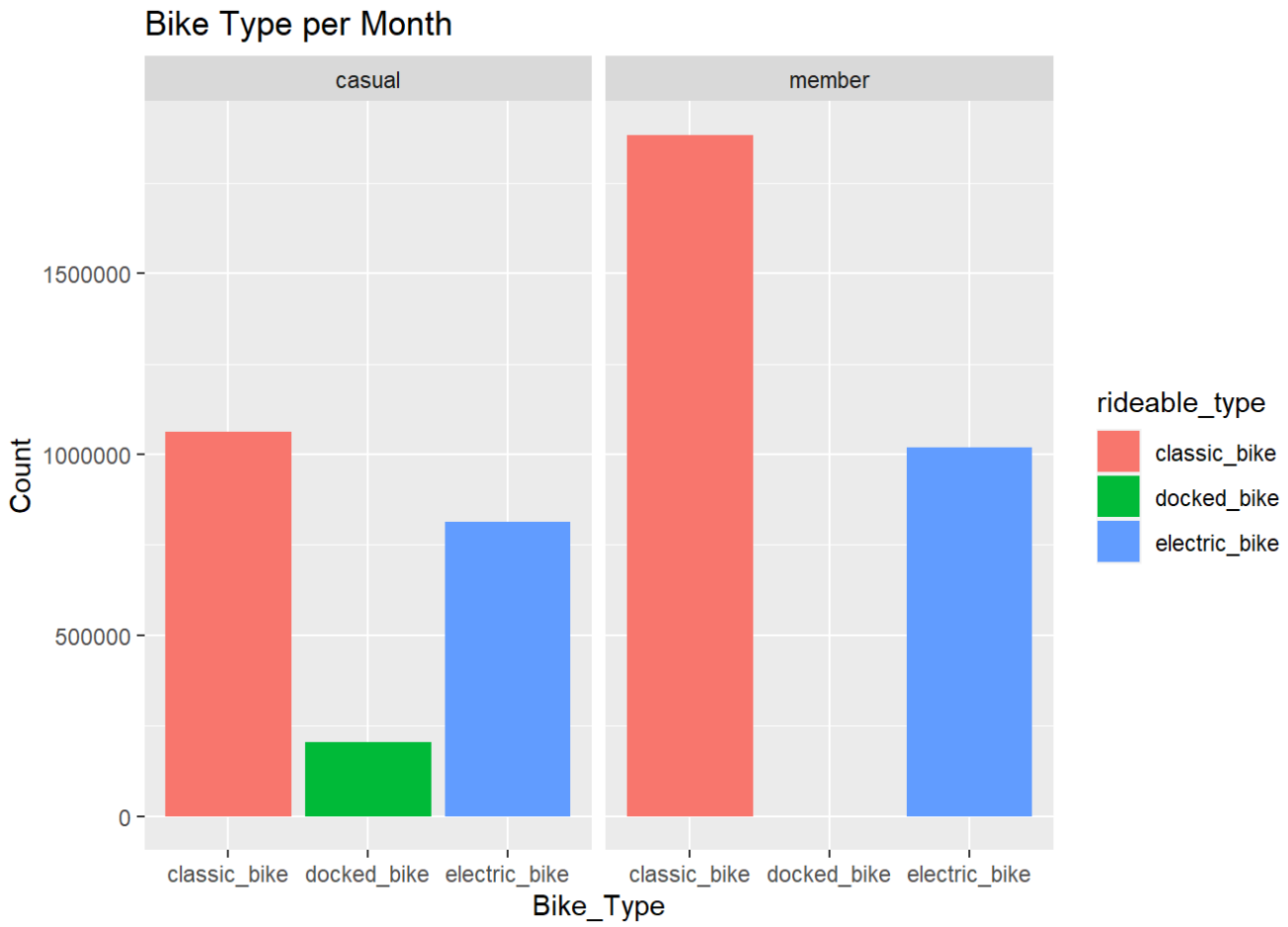
Let us observe the type of bike usage.

```
bike_type_month <- data4 %>%
  select(rideable_type, month) %>%
  group_by(month, rideable_type)%>%
  count() %>%
  arrange()
head(bike_type_month)
```

```
## # A tibble: 6 x 3
## # Groups:   month, rideable_type [6]
##   month rideable_type     n
##   <ord> <chr>         <int>
## 1 Jan   classic_bike  52632
## 2 Jan   docked_bike     774
## 3 Jan   electric_bike 44644
## 4 Feb   classic_bike  56549
## 5 Feb   docked_bike     970
## 6 Feb   electric_bike 51016
```

Let us graph bike type usage.

```
ggplot(data = data3) + geom_bar(mapping = aes(x = rideable_type, fill = rideable_type)) + facet_w
rap(~member_casual) + labs(title = "Bike Type per Month", x = "Bike_Type", y = "Count")
```



Here we take a look at the number of rides per month

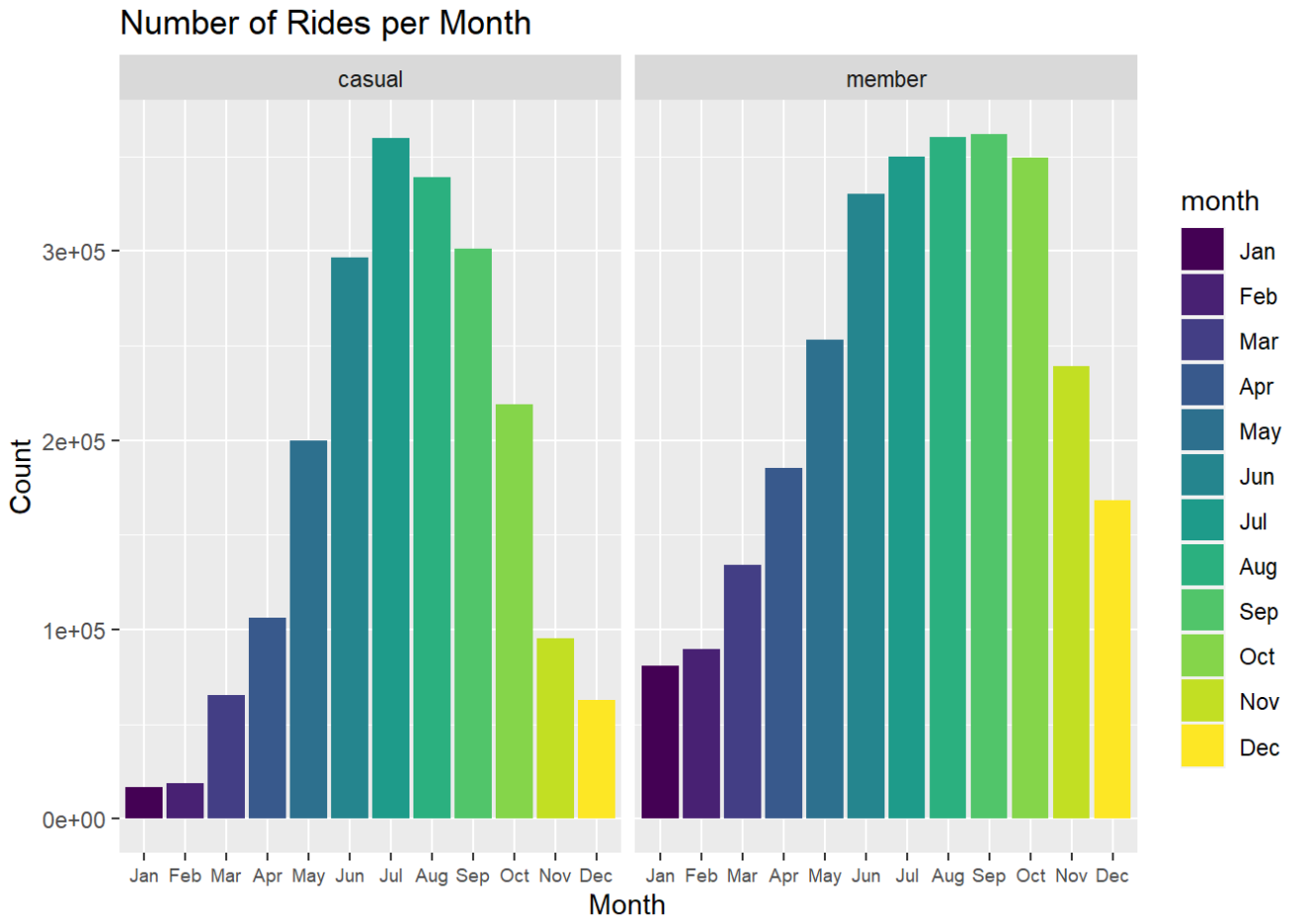
```
rides_per_month_df <- data4 %>%
  select(member_casual , month) %>%
  group_by(month, member_casual)%>%
  count() %>%
  arrange()
head(rides_per_month_df)
```

```
## # A tibble: 6 x 3
## # Groups:   month, member_casual [6]
##   month member_casual      n
##   <ord> <chr>          <int>
## 1 Jan   casual         16867
## 2 Jan   member         81183
## 3 Feb   casual         19027
## 4 Feb   member         89508
## 5 Mar   casual         65197
## 6 Mar   member        134189
```

Let us graph the above chart - Number of Rides per Month.

```
my_ggpp <- ggplot(data = rides_per_month_df,  aes(x = month, y = n,  fill = month))  + geom_col()
+ facet_wrap(~member_casual) + labs(title = "Number of Rides per Month", x = "Month", y = "Count"
)

my_ggpp + theme(axis.text.x = element_text(size = 7))
```



Here we take a look at the number of riders per weekday.

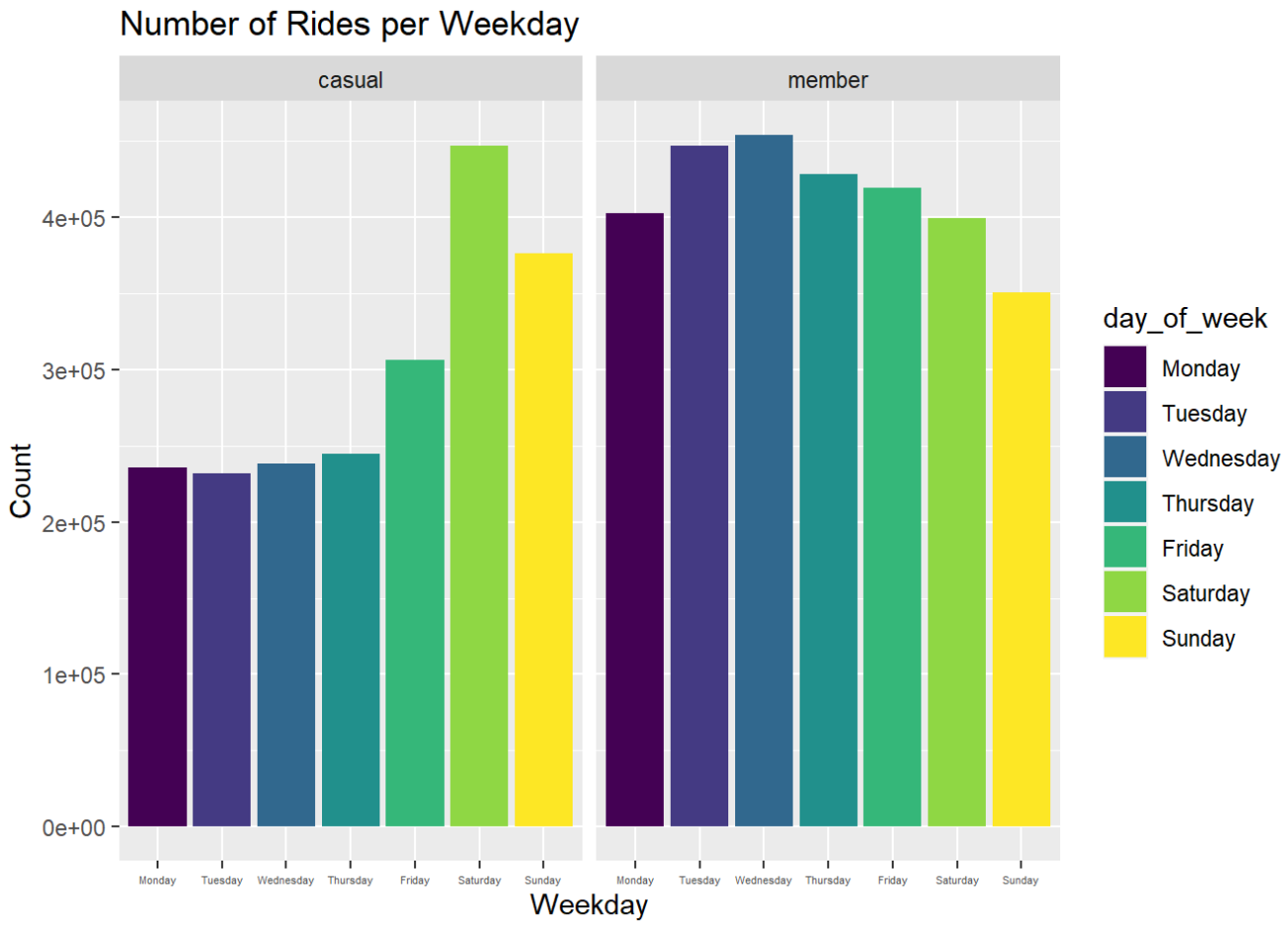
```
data4$day_of_week <- ordered(data4$day_of_week, levels=c("Monday", "Tuesday", "Wednesday", "Thurs
day", "Friday", "Saturday", "Sunday"))

rides_per_week_df <- data4 %>%
  select(member_casual , day_of_week) %>%
  group_by(day_of_week, member_casual)%>%
  count() %>%
  arrange()
head(rides_per_week_df)
```

```
## # A tibble: 6 x 3
## # Groups:   day_of_week, member_casual [6]
##   day_of_week member_casual      n
##   <ord>       <chr>          <int>
## 1 Monday      casual        235744
## 2 Monday      member        402782
## 3 Tuesday     casual        232106
## 4 Tuesday     member        446776
## 5 Wednesday   casual        238216
## 6 Wednesday   member        453784
```

Let us graph the above chart - Number of Rides per Weekday.

```
my_ggp <- ggplot(data = rides_per_week_df,  aes(x = day_of_week, y = n,  fill = day_of_week))  +
geom_col() + facet_wrap(~member_casual) + labs(title = "Number of Rides per Weekday", x = "Weekda
y", y = "Count")
my_ggp + theme(axis.text.x = element_text(size = 4))
```



Let us check the average bicycle held time by each day for members and casual user.

```
data4$day_of_week <- ordered(data4$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesda
y", "Thursday", "Friday", "Saturday"))
```

```
aggregate(data4$held_time2 ~ data4$member_casual + data4$day_of_week, FUN = mean)
```

```
##    data4$member_casual data4$day_of_week data4$held_time2
## 1               casual            Sunday         16.54857
## 2               member            Sunday         12.00184
## 3               casual            Monday         15.16957
## 4               member            Monday         10.51575
## 5               casual           Tuesday         14.26807
## 6               member           Tuesday         10.35744
## 7               casual         Wednesday         14.09343
## 8               member         Wednesday         10.40359
## 9               casual          Thursday         14.01018
## 10              member          Thursday         10.34093
## 11              casual            Friday         14.79868
## 12              member            Friday         10.68165
## 13              casual          Saturday         16.33978
## 14              member          Saturday         11.86852
```

Let us check the average bicycle distance traveled by each day for members and casual user.

```
aggregate(data4$distGeo ~ data4$member_casual + data4$day_of_week, FUN = mean)
```

```
##    data4$member_casual data4$day_of_week data4$distGeo
## 1               casual            Sunday      1.975665
## 2               member            Sunday      1.849581
## 3               casual            Monday      1.857165
## 4               member            Monday      1.739196
## 5               casual           Tuesday      1.890054
## 6               member           Tuesday      1.750280
## 7               casual         Wednesday      1.906968
## 8               member         Wednesday      1.761473
## 9               casual          Thursday      1.911527
## 10              member          Thursday      1.751903
## 11              casual            Friday      1.921687
## 12              member            Friday      1.763484
## 13              casual          Saturday      1.998355
## 14              member          Saturday      1.861595
```

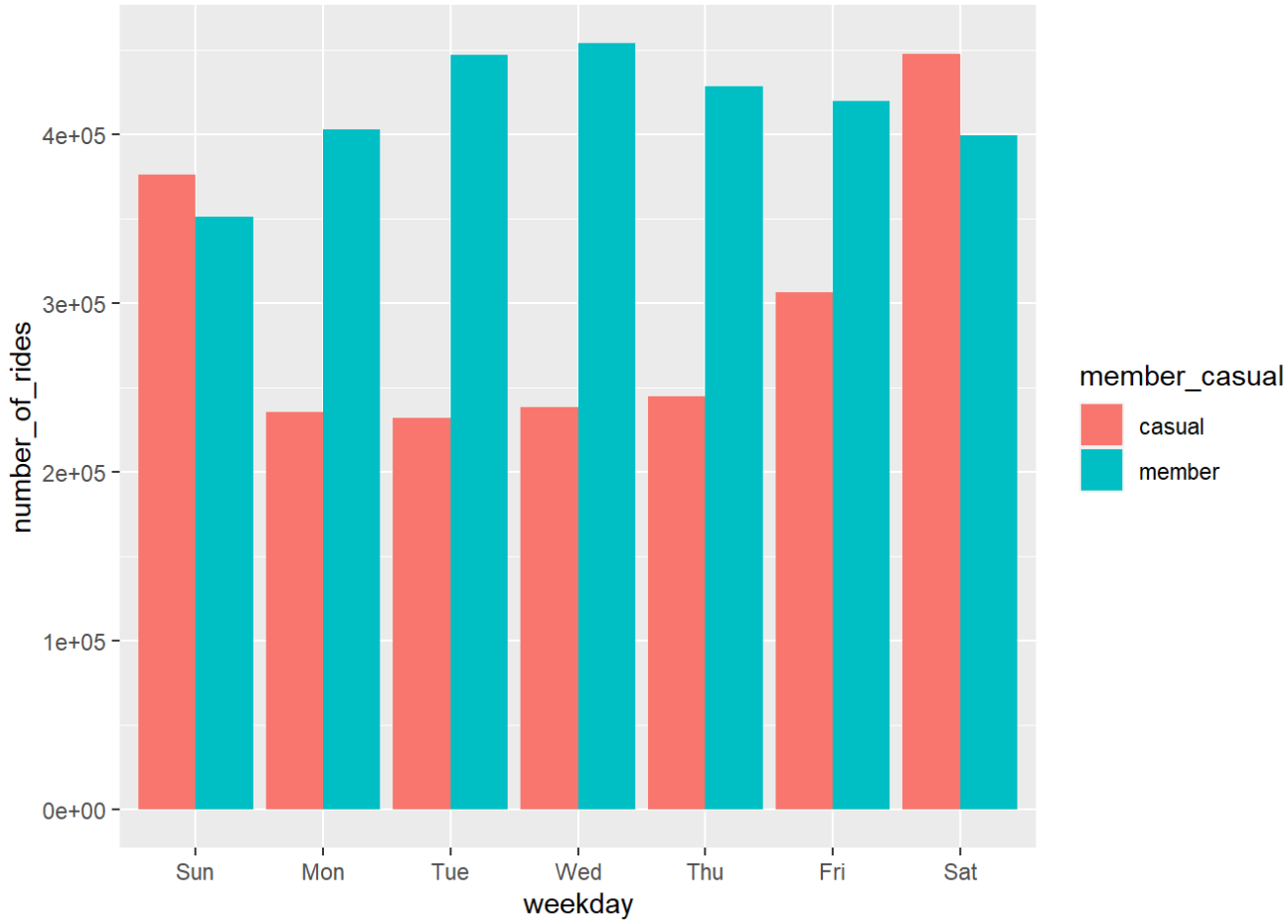
So far, we noticed that casual user utilize the bicycle more than the annual paying members. They held on to the bicycle longer and traveled farther then annual paying members.

```
data4 %>%
  mutate(weekday = wday(start_date, 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 duration
  ,average_duration = mean(held_time2)) %>% # calculates the average duration
  arrange(member_casual, weekday)                        # sorts
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              376255             16.5
##  2 casual        Mon              235744             15.2
##  3 casual        Tue              232106             14.3
##  4 casual        Wed              238216             14.1
##  5 casual        Thu              245020             14.0
##  6 casual        Fri              306111             14.8
##  7 casual        Sat              447211             16.3
##  8 member        Sun              350972             12.0
##  9 member        Mon              402782             10.5
## 10 member        Tue              446776             10.4
## 11 member        Wed              453784             10.4
## 12 member        Thu              428592             10.3
## 13 member        Fri              419575             10.7
## 14 member        Sat              399517             11.9
```

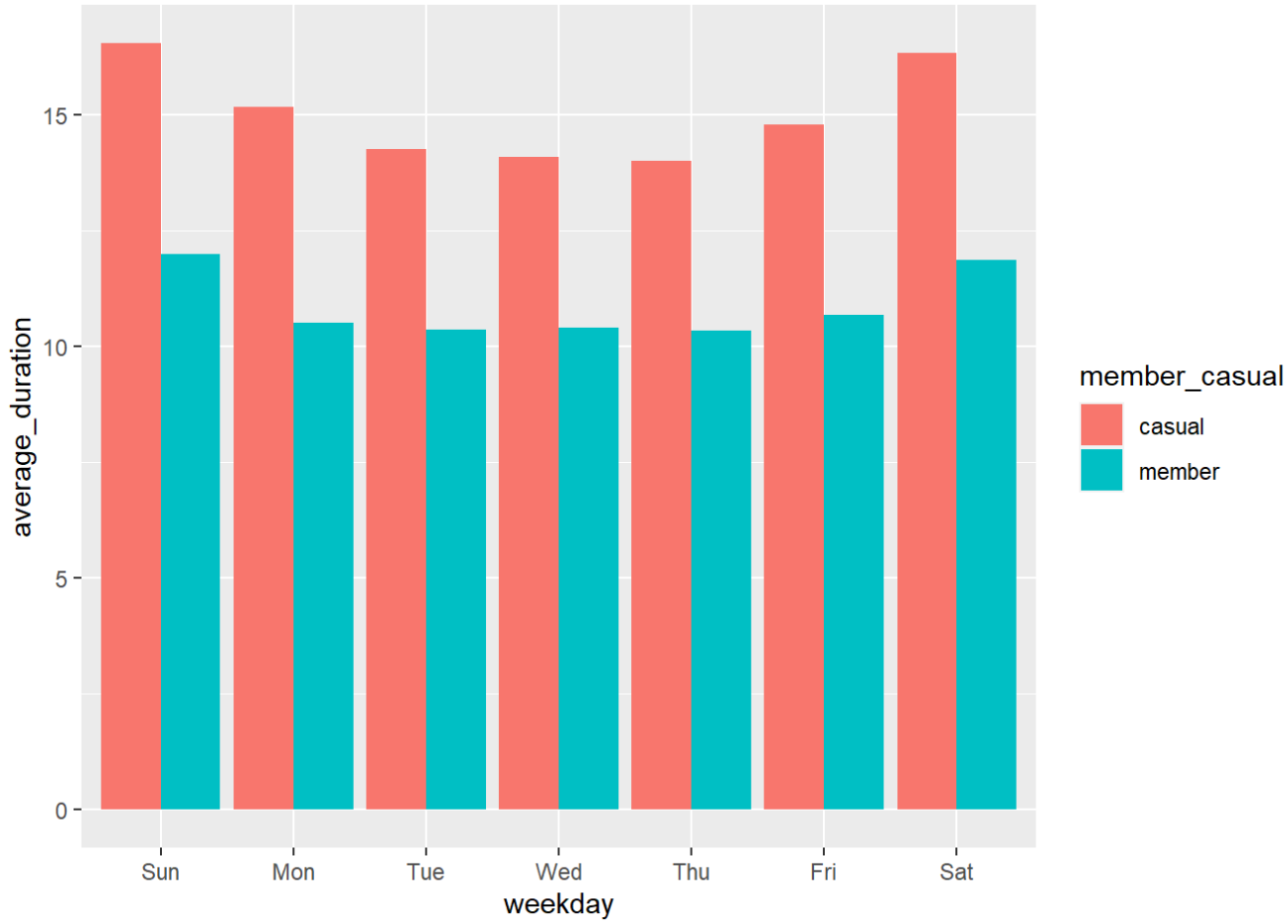
Let us visualize the number of rides by rider type

```
# Let's visualize the number of rides by rider type
data4 %>%
  mutate(weekday = wday(start_date, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(held_time2)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```



Here we look at the average time a casual user and a member held onto the bicycle per day.

```
data4 %>%
  mutate(weekday = wday(start_date, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(held_time2)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```



Just to check our numbers. Looking at aggregates by whole and by member type.

```
data4 %>%
 select(held_time2) %>%
  summarise(max_ride_length = max(held_time2))
```

```
##   max_ride_length
## 1        43.81667
```

```
data4 %>%
  select(held_time2) %>%
  summarise(min_ride_length = min(held_time2))
```

```
##   min_ride_length
## 1      0.01666667
```

```
data4 %>%
  select(member_casual,held_time2) %>%
  summarise(mean_ride_length = mean(held_time2))
```

```
##   mean_ride_length
## 1         12.68268
```

```
df_casual <- data4 %>%
  filter(member_casual == "casual")
```

```
df_casual %>%
  select(held_time2) %>%
summarise(mean_ride_length = mean(held_time2))
```

```
##   mean_ride_length
## 1         15.25559
```

```
df_member_rider <- data4 %>%
  filter(member_casual == "member")
head(df_member_rider)
```

```
##             ride_id rideable_type          started_at            ended_at
## 1: 297268586B79588B  classic_bike 2021-03-20 14:10:41 2021-03-20 14:22:13
## 2: F39301858B6077DD electric_bike 2021-03-23 07:56:51 2021-03-23 08:05:50
## 3: D297F199D875BABE electric_bike 2021-03-31 15:31:19 2021-03-31 15:35:58
## 4: 36B877141175ED7E  classic_bike 2021-03-11 17:37:37 2021-03-11 17:52:44
## 5: 1728D115DBBDF01C  classic_bike 2021-03-13 13:00:02 2021-03-13 13:18:16
## 6: 42179FE11265F287 electric_bike 2021-03-13 10:06:56 2021-03-13 10:22:34
##        start_station_name start_station_id             end_station_name
## 1:   State St & Kinzie St            13050   Lake Shore Dr & North Blvd
## 2:     Shore Dr & 55th St     TA1308000009          Ellis Ave & 60th St
## 3:   Clinton St & Lake St            13021   Franklin St & Jackson Blvd
## 4: Michigan Ave & Lake St     TA1305000011 Racine Ave & Washington Blvd
## 5: Damen Ave & Madison St            13134         Federal St & Polk St
## 6: Damen Ave & Madison St            13134         Federal St & Polk St
##    end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1:         LF-005  41.88919 -87.62775 41.91172 -87.62680        member
## 2:   KA1503000014  41.79523 -87.58083 41.78522 -87.60108        member
## 3:   TA1305000025  41.88555 -87.64173 41.87729 -87.63616        member
## 4:            654  41.88602 -87.62412 41.88307 -87.65695        member
## 5:         SL-008  41.88137 -87.67493 41.87208 -87.62954        member
## 6:         SL-008  41.88141 -87.67490 41.87213 -87.62995        member
##             start_date            end_date Start_Yr Start_Mth Start_Day
## 1: 2021-03-20 14:10:41 2021-03-20 14:22:13     2021         3         7
## 2: 2021-03-23 07:56:51 2021-03-23 08:05:50     2021         3         3
## 3: 2021-03-31 15:31:19 2021-03-31 15:35:58     2021         3         4
## 4: 2021-03-11 17:37:37 2021-03-11 17:52:44     2021         3         5
## 5: 2021-03-13 13:00:02 2021-03-13 13:18:16     2021         3         7
## 6: 2021-03-13 10:06:56 2021-03-13 10:22:34     2021         3         7
##    Start_Hr End_Yr End_Mth End_Day End_Hr held_time held_time2 distHaversine
## 1:       14   2021       3       7     14  692 secs  11.533333         2.510
## 2:        7   2021       3       3      8  539 secs   8.983333         2.017
## 3:       15   2021       3       4     15  279 secs   4.650000         1.029
## 4:       17   2021       3       5     17  907 secs  15.116667         2.741
## 5:       13   2021       3       7     13 1094 secs  18.233333         3.902
## 6:       10   2021       3       7     10  938 secs  15.633333         3.866
##    distVincentyEllipsoid distGeo day_of_week month
## 1:                 2.504   2.504    Saturday   Mar
## 2:                 2.018   2.018     Tuesday   Mar
## 3:                 1.028   1.028   Wednesday   Mar
## 4:                 2.745   2.745    Thursday   Mar
## 5:                 3.906   3.906    Saturday   Mar
## 6:                 3.871   3.871    Saturday   Mar
```

```
df_member_rider %>%
  select(held_time2) %>%
summarise(mean_ride_length = mean(held_time2))
```

```
##   mean_ride_length
## 1         10.83797
```

All numbers matched the aggregate numbers above.

```
counts <- aggregate(data4$held_time2 ~ data4$member_casual + data4$day_of_week, FUN = mean)
```

```
View(counts)
```

```
write.csv(counts,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Mean Time Held Bike - Member v
s. Casual User", row.names = FALSE)
```

```
counts2 <- aggregate(data4$distGeo ~ data4$member_casual + data4$day_of_week, FUN = mean)
```

```
View(counts2)
```

```
write.csv(counts2,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Mean Distance Traveled - Membe
r vs. Casual User", row.names = FALSE)
```

```
write.csv(df_members,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Members and Casual Riders",
row.names = FALSE)
```

```
write.csv(bike_type_month,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Type of Bike Used per
 Month", row.names = FALSE)
```

```
write.csv(rides_per_month_df,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Number of Bike Ride
s Per Month", row.names = FALSE)
```

```
write.csv(rides_per_week_df,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Type of Bike Used pe
r Week", row.names = FALSE)
```

```
write.csv(df_casual,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Casual Members Only", row.na
mes = FALSE)
```

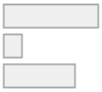
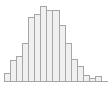
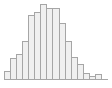
```
print(dfSummary(df_casual, graph.magnif = 0.75), method = 'render')
```

Data Frame Summary

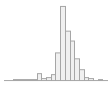
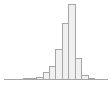
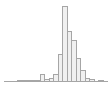
df_casual

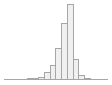
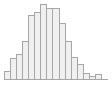
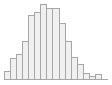
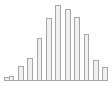
Dimensions: 2080663 x 30
Duplicates: 0

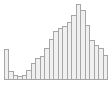
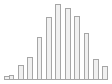
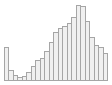
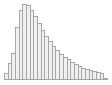
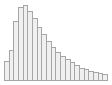
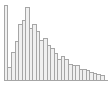
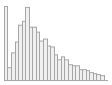
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing

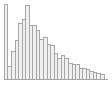
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	ride_id [character]	1. 00001B4F79D102B5 2. 00001DCF2BC423F4 3. 000020C92AA9D6F7 4. 000043B681BFB305 5. 0000453517CABB51 6. 00005B17AF7201F6 7. 00005C84FF78750D 8. 0000664E22910718 9. 000075681ADB8CEB 10. 00007A060B74A702 [2080653 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 2080653 (100.0%)		2080663 (100.0%)	0 (0.0%)
2	rideable_type [character]	1. classic_bike 2. docked_bike 3. electric_bike	1062367 (51.1%) 203734 (9.8%) 814562 (39.1%)		2080663 (100.0%)	0 (0.0%)
3	started_at [POSIXct, POSIXt]	min : 2021-03-01 00:12:56 med : 2021-08-01 19:50:51 max : 2022-02-28 23:57:57 range : 11m 27d 23H 45M 1S	1912189 distinct values		2080663 (100.0%)	0 (0.0%)
4	ended_at [POSIXct, POSIXt]	min : 2021-03-01 00:17:59 med : 2021-08-01 20:06:36 max : 2022-03-01 00:04:15 range : 1y 0m -1d 23H 46M 16S	1907416 distinct values		2080663 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
5	start_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Millennium Park 4. Michigan Ave & Oak St 5. Wells St & Concord Ln 6. Shedd Aquarium 7. Theater on the Lake 8. Wells St & Elm St 9. Clark St & Elm St 10. Clark St & Lincoln Ave [834 others]	274912 (13.2%) 46123 (2.2%) 21713 (1.0%) 19861 (1.0%) 18220 (0.9%) 17691 (0.9%) 16248 (0.8%) 15544 (0.7%) 14818 (0.7%) 14735 (0.7%) 1620798 (77.9%)		2080663 (100.0%)	0 (0.0%)
6	start_station_id [character]	1. (Empty string) 2. 13022 3. 13300 4. LF-005 5. 13008 6. 13042 7. TA1308000050 8. 15544 9. TA1308000001 10. KA1504000135 [824 others]	274910 (13.2%) 46123 (2.2%) 24071 (1.2%) 24001 (1.2%) 21713 (1.0%) 19861 (1.0%) 18220 (0.9%) 17691 (0.9%) 16248 (0.8%) 15544 (0.7%) 1602281 (77.0%)		2080663 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
7	end_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Millennium Park 4. Michigan Ave & Oak St 5. Wells St & Concord Ln 6. Shedd Aquarium 7. Theater on the Lake 8. Wells St & Elm St 9. Clark St & Lincoln Ave 10. DuSable Lake Shore Dr & N [835 others]	311325 (15.0%) 46191 (2.2%) 23344 (1.1%) 20597 (1.0%) 17971 (0.9%) 16517 (0.8%) 16339 (0.8%) 14844 (0.7%) 14656 (0.7%) 14046 (0.7%) 1584833 (76.2%)		2080663 (100.0%)	0 (0.0%)
8	end_station_id [character]	1. (Empty string) 2. 13022 3. LF-005 4. 13008 5. 13300 6. 13042 7. TA1308000050 8. 15544 9. TA1308000001 10. KA1504000135 [826 others]	311325 (15.0%) 46191 (2.2%) 27608 (1.3%) 23344 (1.1%) 20642 (1.0%) 20597 (1.0%) 17971 (0.9%) 16517 (0.8%) 16339 (0.8%) 14844 (0.7%) 1565285 (75.2%)		2080663 (100.0%)	0 (0.0%)
9	start_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 42.1 IQR (CV) : 0 (0)	261285 distinct values		2080663 (100.0%)	0 (0.0%)
10	start_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	249583 distinct values		2080663 (100.0%)	0 (0.0%)
11	end_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 42.1 IQR (CV) : 0 (0)	268333 distinct values		2080663 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
12	end_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	250647 distinct values		2080663 (100.0%)	0 (0.0%)
13	member_casual [character]	1. casual	2080663 (100.0%)		2080663 (100.0%)	0 (0.0%)
14	start_date [POSIXct, POSIXt]	min : 2021-03-01 00:12:56 med : 2021-08-01 19:50:51 max : 2022-02-28 23:57:57 range : 11m 27d 23H 45M 1S	1912189 distinct values		2080663 (100.0%)	0 (0.0%)
15	end_date [POSIXct, POSIXt]	min : 2021-03-01 00:17:59 med : 2021-08-01 20:06:36 max : 2022-03-01 00:04:15 range : 1y 0m -1d 23H 46M 16S	1907416 distinct values		2080663 (100.0%)	0 (0.0%)
16	Start_Yr [numeric]	Min : 2021 Mean : 2021 Max : 2022	2021 : 2044769 (98.3%) 2022 : 35894 (1.7%)		2080663 (100.0%)	0 (0.0%)
17	Start_Mth [numeric]	Mean (sd) : 7.4 (2.2) min ≤ med ≤ max: 1 ≤ 7 ≤ 12 IQR (CV) : 3 (0.3)	12 distinct values		2080663 (100.0%)	0 (0.0%)
18	Start_Day [numeric]	Mean (sd) : 4.2 (2.2) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 376255 (18.1%) 2 : 235744 (11.3%) 3 : 232106 (11.2%) 4 : 238216 (11.4%) 5 : 245020 (11.8%) 6 : 306111 (14.7%) 7 : 447211 (21.5%)		2080663 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
19	Start_Hr [integer]	Mean (sd) : 14.6 (5.3) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 6 (0.4)	24 distinct values		2080663 (100.0%)	0 (0.0%)
20	End_Yr [numeric]	Min : 2021 Mean : 2021 Max : 2022	2021 : 2044753 (98.3%) 2022 : 35910 (1.7%)		2080663 (100.0%)	0 (0.0%)
21	End_Mth [numeric]	Mean (sd) : 7.4 (2.2) min ≤ med ≤ max: 1 ≤ 7 ≤ 12 IQR (CV) : 3 (0.3)	12 distinct values		2080663 (100.0%)	0 (0.0%)
22	End_Day [numeric]	Mean (sd) : 4.2 (2.2) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 379101 (18.2%) 2 : 236188 (11.4%) 3 : 232087 (11.2%) 4 : 238072 (11.4%) 5 : 244524 (11.8%) 6 : 304594 (14.6%) 7 : 446097 (21.4%)		2080663 (100.0%)	0 (0.0%)
23	End_Hr [integer]	Mean (sd) : 14.7 (5.4) min ≤ med ≤ max: 0 ≤ 16 ≤ 23 IQR (CV) : 6 (0.4)	24 distinct values		2080663 (100.0%)	0 (0.0%)
24	held_time [difftime]	min : 1 med : 784 max : 2629 units : secs	2629 distinct values		2080663 (100.0%)	0 (0.0%)
25	held_time2 [numeric]	Mean (sd) : 15.3 (9.6) min ≤ med ≤ max: 0 ≤ 13.1 ≤ 43.8 IQR (CV) : 12.8 (0.6)	2629 distinct values		2080663 (100.0%)	0 (0.0%)
26	distHaversine [numeric]	Mean (sd) : 1.9 (1.3) min ≤ med ≤ max: 0 ≤ 1.7 ≤ 5.6 IQR (CV) : 1.7 (0.7)	5626 distinct values		2080663 (100.0%)	0 (0.0%)
27	distVincentyEllipsoid [numeric]	Mean (sd) : 1.9 (1.3) min ≤ med ≤ max: 0 ≤ 1.7 ≤ 5.6 IQR (CV) : 1.7 (0.7)	5613 distinct values		2080663 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
28	distGeo [numeric]	Mean (sd) : 1.9 (1.3) min ≤ med ≤ max: 0 ≤ 1.7 ≤ 5.6 IQR (CV) : 1.7 (0.7)	5613 distinct values		2080663 (100.0%)	0 (0.0%)
29	day_of_week [ordered, factor]	1. Sunday 2. Monday 3. Tuesday 4. Wednesday 5. Thursday 6. Friday 7. Saturday	376255 (18.1%) 235744 (11.3%) 232106 (11.2%) 238216 (11.4%) 245020 (11.8%) 306111 (14.7%) 447211 (21.5%)		2080663 (100.0%)	0 (0.0%)
30	month [ordered, factor]	1. Jan 2. Feb 3. Mar 4. Apr 5. May 6. Jun 7. Jul 8. Aug 9. Sep 10. Oct [2 others]	16867 (0.8%) 19027 (0.9%) 65197 (3.1%) 106257 (5.1%) 199912 (9.6%) 296705 (14.3%) 359613 (17.3%) 339049 (16.3%) 301051 (14.5%) 219090 (10.5%) 157895 (7.6%)		2080663 (100.0%)	0 (0.0%)

Generated by summarytools (https://github.com/dcomtois/summarytools) 1.0.1 (R (https://www.r-project.org/) version 4.1.2)
2022-06-21

```
write.csv(df_member_rider,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Annual Members Only",
 row.names = FALSE)
```

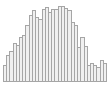
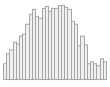
```
print(dfSummary(df_member_rider, graph.magnif = 0.75), method = 'render')
```

Data Frame Summary

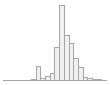
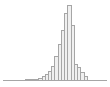
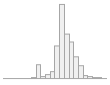
df_member_rider

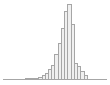
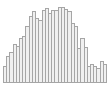
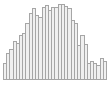
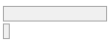
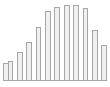
Dimensions: 2901998 x 30
Duplicates: 0

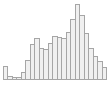
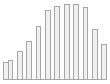
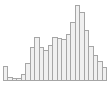
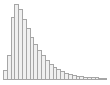
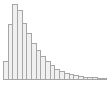
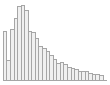
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing

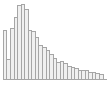
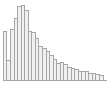
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	ride_id [character]	1. 00000123F60251E6 2. 000002EBE159AE82 3. 00000B4F1F71F9C2 4. 00000EBBC119168C 5. 000019B7F053D461 6. 00001A81D056B01B 7. 00001BEE76AB24E0 8. 0000228A4B430869 9. 000025C113FEB7B6 10. 0000278F02EFFEF9 [2901988 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 2901988 (100.0%)		2901998 (100.0%)	0 (0.0%)
2	rideable_type [character]	1. classic_bike 2. electric_bike	1882570 (64.9%) 1019428 (35.1%)		2901998 (100.0%)	0 (0.0%)
3	started_at [POSIXct, POSIXt]	min : 2021-03-01 00:05:42 med : 2021-08-18 08:27:43 max : 2022-02-28 23:58:38 range : 11m 27d 23H 52M 56S	2662265 distinct values		2901998 (100.0%)	0 (0.0%)
4	ended_at [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-18 08:37:10 max : 2022-03-01 00:20:28 range : 1y 0m 0d 0H 14M 0S	2660447 distinct values		2901998 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
5	start_station_name [character]	1. (Empty string) 2. Kingsbury St & Kinzie St 3. Clark St & Elm St 4. Wells St & Concord Ln 5. Wells St & Elm St 6. Wells St & Huron St 7. Dearborn St & Erie St 8. St. Clair St & Erie St 9. Clinton St & Madison St 10. Broadway & Barry Ave [830 others]	350048 (12.1%) 23939 (0.8%) 23808 (0.8%) 23287 (0.8%) 20604 (0.7%) 18556 (0.6%) 18484 (0.6%) 17668 (0.6%) 16828 (0.6%) 16695 (0.6%) 2372081 (81.7%)		2901998 (100.0%)	0 (0.0%)
6	start_station_id [character]	1. (Empty string) 2. KA1503000043 3. TA1307000039 4. TA1308000050 5. KA1504000135 6. TA1306000012 7. 13045 8. 13016 9. TA1305000032 10. 13137 [821 others]	350048 (12.1%) 23939 (0.8%) 23808 (0.8%) 23287 (0.8%) 20604 (0.7%) 18556 (0.6%) 18484 (0.6%) 17668 (0.6%) 16828 (0.6%) 16695 (0.6%) 2372081 (81.7%)		2901998 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
7	end_station_name [character]	1. (Empty string) 2. Clark St & Elm St 3. Kingsbury St & Kinzie St 4. Wells St & Concord Ln 5. Wells St & Elm St 6. Dearborn St & Erie St 7. Wells St & Huron St 8. St. Clair St & Erie St 9. Clinton St & Madison St 10. Broadway & Barry Ave [823 others]	353154 (12.2%) 23991 (0.8%) 23935 (0.8%) 23895 (0.8%) 21360 (0.7%) 19073 (0.7%) 18296 (0.6%) 17984 (0.6%) 17357 (0.6%) 17327 (0.6%) 2365626 (81.5%)		2901998 (100.0%)	0 (0.0%)
8	end_station_id [character]	1. (Empty string) 2. TA1307000039 3. KA1503000043 4. TA1308000050 5. KA1504000135 6. 13045 7. TA1306000012 8. 13016 9. LF-005 10. TA1305000032 [814 others]	353154 (12.2%) 23991 (0.8%) 23935 (0.8%) 23895 (0.8%) 21360 (0.7%) 19073 (0.7%) 18296 (0.6%) 17984 (0.6%) 17486 (0.6%) 17357 (0.6%) 2365467 (81.5%)		2901998 (100.0%)	0 (0.0%)
9	start_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 42.1 IQR (CV) : 0 (0)	305299 distinct values		2901998 (100.0%)	0 (0.0%)
10	start_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	293742 distinct values		2901998 (100.0%)	0 (0.0%)
11	end_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 42.1 IQR (CV) : 0 (0)	312570 distinct values		2901998 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
12	end_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	288668 distinct values		2901998 (100.0%)	0 (0.0%)
13	member_casual [character]	1. member	2901998 (100.0%)		2901998 (100.0%)	0 (0.0%)
14	start_date [POSIXct, POSIXt]	min : 2021-03-01 00:05:42 med : 2021-08-18 08:27:43 max : 2022-02-28 23:58:38 range : 11m 27d 23H 52M 56S	2662265 distinct values		2901998 (100.0%)	0 (0.0%)
15	end_date [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-18 08:37:10 max : 2022-03-01 00:20:28 range : 1y 0m 0d 0H 14M 0S	2660447 distinct values		2901998 (100.0%)	0 (0.0%)
16	Start_Yr [numeric]	Min : 2021 Mean : 2021.1 Max : 2022	2021 : 2731307 (94.1%) 2022 : 170691 (5.9%)		2901998 (100.0%)	0 (0.0%)
17	Start_Mth [numeric]	Mean (sd) : 7.4 (2.8) min ≤ med ≤ max: 1 ≤ 8 ≤ 12 IQR (CV) : 5 (0.4)	12 distinct values		2901998 (100.0%)	0 (0.0%)
18	Start_Day [numeric]	Mean (sd) : 4.1 (1.9) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 350972 (12.1%) 2 : 402782 (13.9%) 3 : 446776 (15.4%) 4 : 453784 (15.6%) 5 : 428592 (14.8%) 6 : 419575 (14.5%) 7 : 399517 (13.8%)		2901998 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
19	Start_Hr [integer]	Mean (sd) : 14 (4.9) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 8 (0.4)	24 distinct values		2901998 (100.0%)	0 (0.0%)
20	End_Yr [numeric]	Min : 2021 Mean : 2021.1 Max : 2022	2021 : 2731297 (94.1%) 2022 : 170701 (5.9%)		2901998 (100.0%)	0 (0.0%)
21	End_Mth [numeric]	Mean (sd) : 7.4 (2.8) min ≤ med ≤ max: 1 ≤ 8 ≤ 12 IQR (CV) : 5 (0.4)	12 distinct values		2901998 (100.0%)	0 (0.0%)
22	End_Day [numeric]	Mean (sd) : 4.1 (1.9) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 352260 (12.1%) 2 : 402832 (13.9%) 3 : 446635 (15.4%) 4 : 453699 (15.6%) 5 : 428384 (14.8%) 6 : 418941 (14.4%) 7 : 399247 (13.8%)		2901998 (100.0%)	0 (0.0%)
23	End_Hr [integer]	Mean (sd) : 14.1 (5) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 7 (0.4)	24 distinct values		2901998 (100.0%)	0 (0.0%)
24	held_time [difftime]	min : 1 med : 529 max : 2629 units : secs	2629 distinct values		2901998 (100.0%)	0 (0.0%)
25	held_time2 [numeric]	Mean (sd) : 10.8 (7.7) min ≤ med ≤ max: 0 ≤ 8.8 ≤ 43.8 IQR (CV) : 9.1 (0.7)	2629 distinct values		2901998 (100.0%)	0 (0.0%)
26	distHaversine [numeric]	Mean (sd) : 1.8 (1.3) min ≤ med ≤ max: 0 ≤ 1.4 ≤ 5.6 IQR (CV) : 1.6 (0.7)	5626 distinct values		2901998 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
27	distVincentyEllipsoid [numeric]	Mean (sd) : 1.8 (1.3) min ≤ med ≤ max: 0 ≤ 1.4 ≤ 5.6 IQR (CV) : 1.6 (0.7)	5613 distinct values		2901998 (100.0%)	0 (0.0%)
28	distGeo [numeric]	Mean (sd) : 1.8 (1.3) min ≤ med ≤ max: 0 ≤ 1.4 ≤ 5.6 IQR (CV) : 1.6 (0.7)	5613 distinct values		2901998 (100.0%)	0 (0.0%)
29	day_of_week [ordered, factor]	1. Sunday 2. Monday 3. Tuesday 4. Wednesday 5. Thursday 6. Friday 7. Saturday	350972 (12.1%) 402782 (13.9%) 446776 (15.4%) 453784 (15.6%) 428592 (14.8%) 419575 (14.5%) 399517 (13.8%)		2901998 (100.0%)	0 (0.0%)
30	month [ordered, factor]	1. Jan 2. Feb 3. Mar 4. Apr 5. May 6. Jun 7. Jul 8. Aug 9. Sep 10. Oct [2 others]	81183 (2.8%) 89508 (3.1%) 134189 (4.6%) 185231 (6.4%) 253135 (8.7%) 330361 (11.4%) 349748 (12.1%) 360318 (12.4%) 361760 (12.5%) 349285 (12.0%) 407280 (14.0%)		2901998 (100.0%)	0 (0.0%)

Generated by summarytools (https://github.com/dcomtois/summarytools) 1.0.1 (R (https://www.r-project.org/) version 4.1.2)
2022-06-21

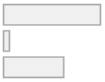
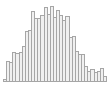
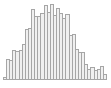
```
write.csv(data4,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Bike-Share Data Complete", row.names = FALSE)
```

```
print(dfSummary(data4, graph.magnif = 0.75), method = 'render')
```

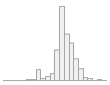
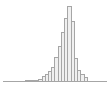
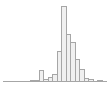
Data Frame Summary

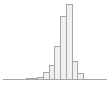
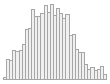
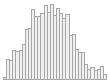
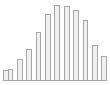
data4

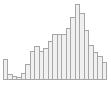
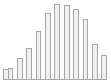
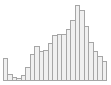
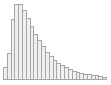
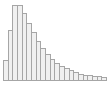
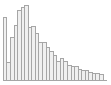
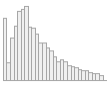
Dimensions: 4982661 x 30
Duplicates: 0

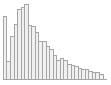
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	ride_id [character]	1. 00000123F60251E6 2. 000002EBE159AE82 3. 00000B4F1F71F9C2 4. 00000EBBC119168C 5. 000019B7F053D461 6. 00001A81D056B01B 7. 00001B4F79D102B5 8. 00001BEE76AB24E0 9. 00001DCF2BC423F4 10. 000020C92AA9D6F7 [4982651 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 4982651 (100.0%)		4982661 (100.0%)	0 (0.0%)
2	rideable_type [character]	1. classic_bike 2. docked_bike 3. electric_bike	2944937 (59.1%) 203734 (4.1%) 1833990 (36.8%)		4982661 (100.0%)	0 (0.0%)
3	started_at [POSIXct, POSIXt]	min : 2021-03-01 00:05:42 med : 2021-08-10 13:20:54 max : 2022-02-28 23:58:38 range : 11m 27d 23H 52M 56S	4271443 distinct values		4982661 (100.0%)	0 (0.0%)
4	ended_at [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-10 13:32:39 max : 2022-03-01 00:20:28 range : 1y 0m 0d 0H 14M 0S	4266002 distinct values		4982661 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
5	start_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Wells St & Concord Ln 4. Clark St & Elm St 5. Wells St & Elm St 6. Kingsbury St & Kinzie St 7. Michigan Ave & Oak St 8. Clark St & Lincoln Ave 9. Clark St & Armitage Ave 10. Wells St & Huron St [843 others]	624960 (12.5%) 59655 (1.2%) 41507 (0.8%) 38626 (0.8%) 36148 (0.7%) 33266 (0.7%) 31899 (0.6%) 30376 (0.6%) 29897 (0.6%) 29677 (0.6%) 4026650 (80.8%)		4982661 (100.0%)	0 (0.0%)
6	start_station_id [character]	1. (Empty string) 2. 13022 3. TA1308000050 4. LF-005 5. TA1307000039 6. KA1504000135 7. KA1503000043 8. 13300 9. 13042 10. 13179 [834 others]	624958 (12.5%) 59655 (1.2%) 41507 (0.8%) 40476 (0.8%) 38626 (0.8%) 36148 (0.7%) 33266 (0.7%) 32477 (0.7%) 31899 (0.6%) 30376 (0.6%) 4013273 (80.5%)		4982661 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
7	end_station_name [character]	1. (Empty string) 2. Streeter Dr & Grand Ave 3. Wells St & Concord Ln 4. Clark St & Elm St 5. Wells St & Elm St 6. Kingsbury St & Kinzie St 7. Michigan Ave & Oak St 8. Millennium Park 9. Clark St & Lincoln Ave 10. Dearborn St & Erie St [844 others]	664479 (13.3%) 58287 (1.2%) 41866 (0.8%) 38031 (0.8%) 36204 (0.7%) 32732 (0.7%) 31826 (0.6%) 30621 (0.6%) 30067 (0.6%) 29653 (0.6%) 3988895 (80.1%)		4982661 (100.0%)	0 (0.0%)
8	end_station_id [character]	1. (Empty string) 2. 13022 3. LF-005 4. TA1308000050 5. TA1307000039 6. KA1504000135 7. KA1503000043 8. 13042 9. 13008 10. 13179 [836 others]	664479 (13.3%) 58287 (1.2%) 45094 (0.9%) 41866 (0.8%) 38031 (0.8%) 36204 (0.7%) 32732 (0.7%) 31826 (0.6%) 30621 (0.6%) 30067 (0.6%) 3973454 (79.7%)		4982661 (100.0%)	0 (0.0%)
9	start_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 42.1 IQR (CV) : 0 (0)	390381 distinct values		4982661 (100.0%)	0 (0.0%)
10	start_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	371411 distinct values		4982661 (100.0%)	0 (0.0%)
11	end_lat [numeric]	Mean (sd) : 41.9 (0) min ≤ med ≤ max: 41.6 ≤ 41.9 ≤ 42.1 IQR (CV) : 0 (0)	412613 distinct values		4982661 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
12	end_lng [numeric]	Mean (sd) : -87.6 (0) min ≤ med ≤ max: -87.8 ≤ -87.6 ≤ -87.5 IQR (CV) : 0 (0)	374985 distinct values		4982661 (100.0%)	0 (0.0%)
13	member_casual [character]	1. casual 2. member	2080663 (41.8%) 2901998 (58.2%)		4982661 (100.0%)	0 (0.0%)
14	start_date [POSIXct, POSIXt]	min : 2021-03-01 00:05:42 med : 2021-08-10 13:20:54 max : 2022-02-28 23:58:38 range : 11m 27d 23H 52M 56S	4271443 distinct values		4982661 (100.0%)	0 (0.0%)
15	end_date [POSIXct, POSIXt]	min : 2021-03-01 00:06:28 med : 2021-08-10 13:32:39 max : 2022-03-01 00:20:28 range : 1y 0m 0d 0H 14M 0S	4266002 distinct values		4982661 (100.0%)	0 (0.0%)
16	Start_Yr [numeric]	Min : 2021 Mean : 2021 Max : 2022	2021 : 4776076 (95.9%) 2022 : 206585 (4.1%)		4982661 (100.0%)	0 (0.0%)
17	Start_Mth [numeric]	Mean (sd) : 7.4 (2.6) min ≤ med ≤ max: 1 ≤ 8 ≤ 12 IQR (CV) : 3 (0.4)	12 distinct values		4982661 (100.0%)	0 (0.0%)
18	Start_Day [numeric]	Mean (sd) : 4.1 (2) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 727227 (14.6%) 2 : 638526 (12.8%) 3 : 678882 (13.6%) 4 : 692000 (13.9%) 5 : 673612 (13.5%) 6 : 725686 (14.6%) 7 : 846728 (17.0%)		4982661 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
19	Start_Hr [integer]	Mean (sd) : 14.2 (5.1) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 7 (0.4)	24 distinct values		4982661 (100.0%)	0 (0.0%)
20	End_Yr [numeric]	Min : 2021 Mean : 2021 Max : 2022	2021 : 4776050 (95.9%) 2022 : 206611 (4.1%)		4982661 (100.0%)	0 (0.0%)
21	End_Mth [numeric]	Mean (sd) : 7.4 (2.6) min ≤ med ≤ max: 1 ≤ 8 ≤ 12 IQR (CV) : 3 (0.4)	12 distinct values		4982661 (100.0%)	0 (0.0%)
22	End_Day [numeric]	Mean (sd) : 4.1 (2.1) min ≤ med ≤ max: 1 ≤ 4 ≤ 7 IQR (CV) : 4 (0.5)	1 : 731361 (14.7%) 2 : 639020 (12.8%) 3 : 678722 (13.6%) 4 : 691771 (13.9%) 5 : 672908 (13.5%) 6 : 723535 (14.5%) 7 : 845344 (17.0%)		4982661 (100.0%)	0 (0.0%)
23	End_Hr [integer]	Mean (sd) : 14.3 (5.1) min ≤ med ≤ max: 0 ≤ 15 ≤ 23 IQR (CV) : 7 (0.4)	24 distinct values		4982661 (100.0%)	0 (0.0%)
24	held_time [difftime]	min : 1 med : 625 max : 2629 units : secs	2629 distinct values		4982661 (100.0%)	0 (0.0%)
25	held_time2 [numeric]	Mean (sd) : 12.7 (8.8) min ≤ med ≤ max: 0 ≤ 10.4 ≤ 43.8 IQR (CV) : 10.9 (0.7)	2629 distinct values		4982661 (100.0%)	0 (0.0%)
26	distHaversine [numeric]	Mean (sd) : 1.8 (1.3) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 5.6 IQR (CV) : 1.7 (0.7)	5626 distinct values		4982661 (100.0%)	0 (0.0%)
27	distVincentyEllipsoid [numeric]	Mean (sd) : 1.8 (1.3) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 5.6 IQR (CV) : 1.7 (0.7)	5613 distinct values		4982661 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
28	distGeo [numeric]	Mean (sd) : 1.8 (1.3) min ≤ med ≤ max: 0 ≤ 1.6 ≤ 5.6 IQR (CV) : 1.7 (0.7)	5613 distinct values		4982661 (100.0%)	0 (0.0%)
29	day_of_week [ordered, factor]	1. Sunday 2. Monday 3. Tuesday 4. Wednesday 5. Thursday 6. Friday 7. Saturday	727227 (14.6%) 638526 (12.8%) 678882 (13.6%) 692000 (13.9%) 673612 (13.5%) 725686 (14.6%) 846728 (17.0%)		4982661 (100.0%)	0 (0.0%)
30	month [ordered, factor]	1. Jan 2. Feb 3. Mar 4. Apr 5. May 6. Jun 7. Jul 8. Aug 9. Sep 10. Oct [2 others]	98050 (2.0%) 108535 (2.2%) 199386 (4.0%) 291488 (5.9%) 453047 (9.1%) 627066 (12.6%) 709361 (14.2%) 699367 (14.0%) 662811 (13.3%) 568375 (11.4%) 565175 (11.3%)		4982661 (100.0%)	0 (0.0%)

Generated by summarytools (https://github.com/dcomtois/summarytools) 1.0.1 (R (https://www.r-project.org/) version 4.1.2)
2022-06-21

```
counts3 <- aggregate(data4$held_time2 ~ data4$member_casual + data4$day_of_week + data4$rideable_type, FUN = mean)
```

```
View(counts3)
```

```
write.csv(counts3,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Type of Bike Used per Week per Bike Type per User Type", row.names = FALSE)
```

```
counts4 <- aggregate(data4$distGeo ~ data4$member_casual + data4$month, FUN = mean)
```

```
View(counts4)
```

```
write.csv(counts4,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Mean Distance Traveled via Bike Used by Member", row.names = FALSE)
```

```
counts5 <- aggregate(data4$held_time2 ~ data4$member_casual + data4$month, FUN = mean)
```

```
View(counts5)
```

```
write.csv(counts5,"C:\\Users\\rodne\\Desktop\\Cyclistic_Bike_Data\\Mean Time Bike Used by Member"
, row.names = FALSE)
```

Share

Charts

The following is the list of charts that will be made in Excel for a Powerpoint presentation. The charts are listed in the order they would be in if presented to the stakeholders.

• Total number of Bike-share users, members, and casual users together (place in the title of the following chart)

• Total number of Bike-share users, by type, members, and casual users (pie chart)

• Total Bike-Share for the past 12 months (place in the following bar chart)

• Total Bike-Share by month for the past 12 months (bar chart)

• Total Bike-Share by week for the past 12 months (bar chart)

• Total number of bike types used (electric, docked, and classic) (bar chart)

• Total number of bike types used (electric, docked, and classic) by month (bar chart)

• Total number of bike types used (electric, docked, and classic) by user type (member and casual) (bar chart)

• Average time (mean time) bikes were used a year (place in the title of the following chart)

• Average time (mean time) bikes were used by month and by user type (member and casual) (bar chart)

• Average time (mean time) bikes were used by week and by user type (member and casual) (bar chart)

• Average distance (mean distance) bikes were used a year (place in the title of the following chart)

• Average distance (mean distance) bikes were used by month (bar chart)

• Average distance (mean distance) bikes were used by month and by user type (member and casual) (bar chart)

We do not want to create charts with cumulative time and distance because there are more members than casual users. The sum will favor the members. In this case, we want to use averages (means).

We now need to determine the working directory in R to store our slide images.

```
getwd()
```

```
## [1] "C:/Users/rodne/OneDrive/Documents"
```

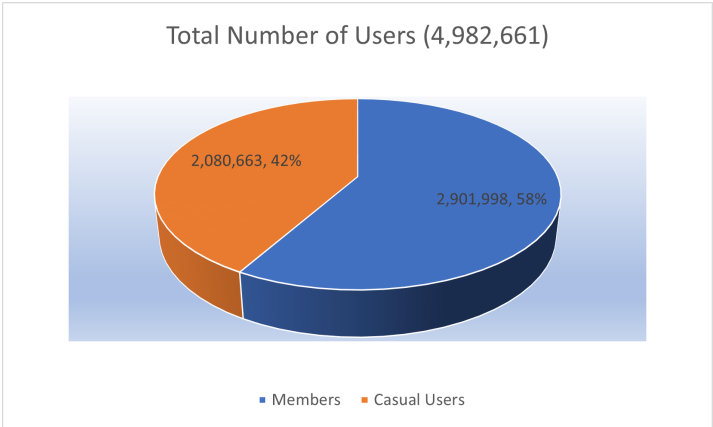
```
knitr::include_graphics('images/hex-rmarkdown.png')
```

Test Image

Now we can begin to add our presentation. All of the following slides were created in Mircosoft Excel.

Total Number of Cyclistic Bike-Share Users

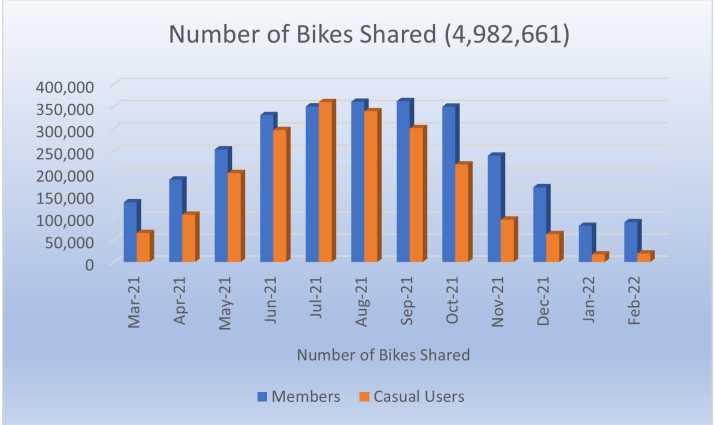
```
knitr::include_graphics('images/Slide_1.png')
```



Total Number of Bike-Share Users

Total Number of Cyclistic Bike-Share Users by Month

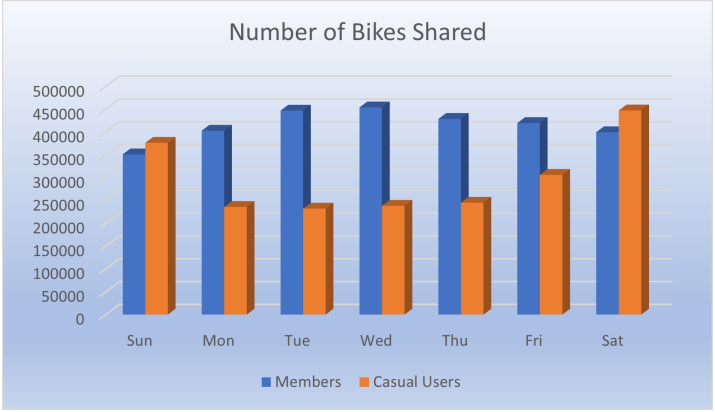
```
knitr::include_graphics('images/Slide_2.png')
```



Total Number of Bike-Share Users per the last 12 Months

Total Number of Cyclistic Bike-Share Users per Weekday

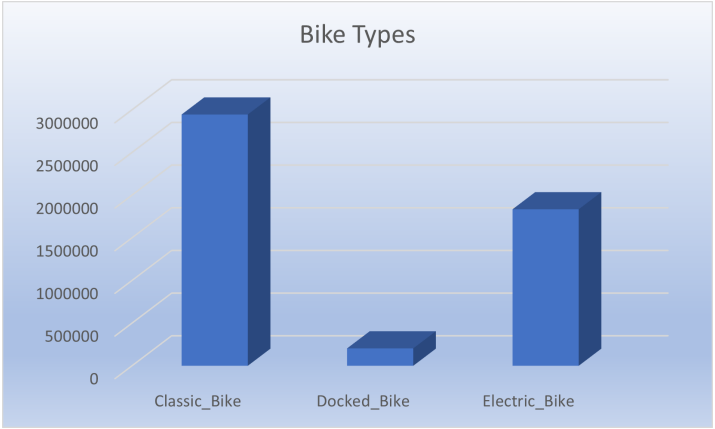
```
knitr::include_graphics('images/Slide_3.png')
```



Total Number of Bike-Share Users per Weekday

Total Number of Cyclistic Bike-Share Bicycle Types

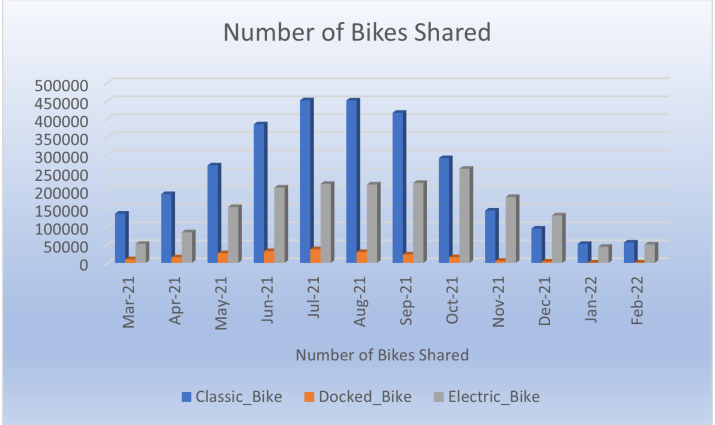
```
knitr::include_graphics('images/Slide_4.png')
```



Total Number of Bike-Share Bicycle Types

Total Number of Cyclistic Bike-Share Bicycle Types by Month

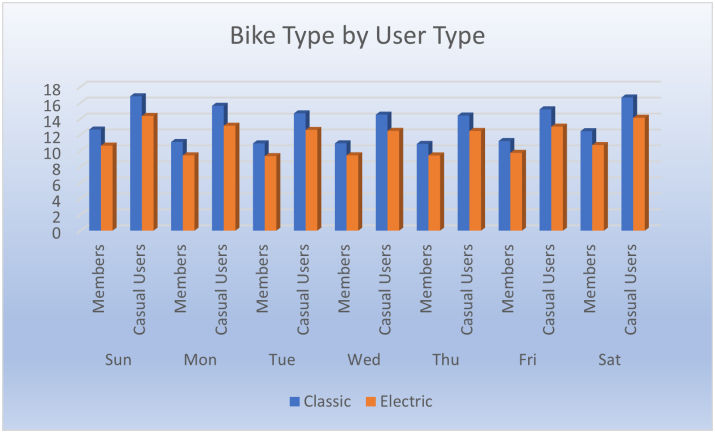
```
knitr::include_graphics('images/Slide_5.png')
```



Total Number of Bike-Share Bicycle Types by Month

Total Number of Cyclistic Bike-Share Bicycle Types by Member Type and by Weekday

```
knitr::include_graphics('images/Slide_6.png')
```

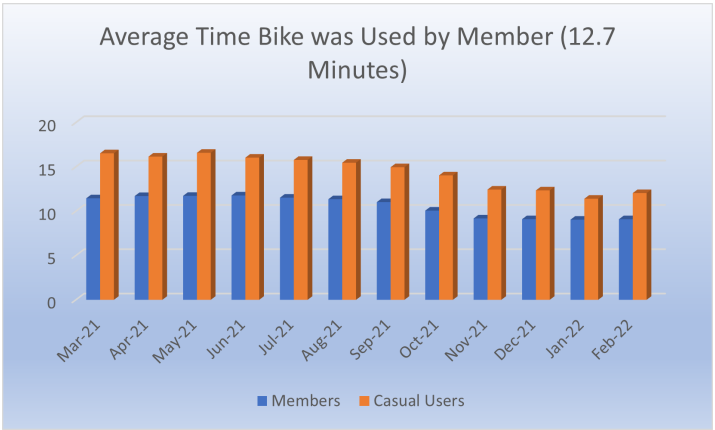


Total Number of Bike-Share Bicycle Types by Member Type and by Weekday

Note, in the above chart, Docked bicycles were removed. The data showed that they were only utilized by Casual Users.

Average time (mean) Cyclistic Bike-Share Bicycles were Used by Month and by Member Type

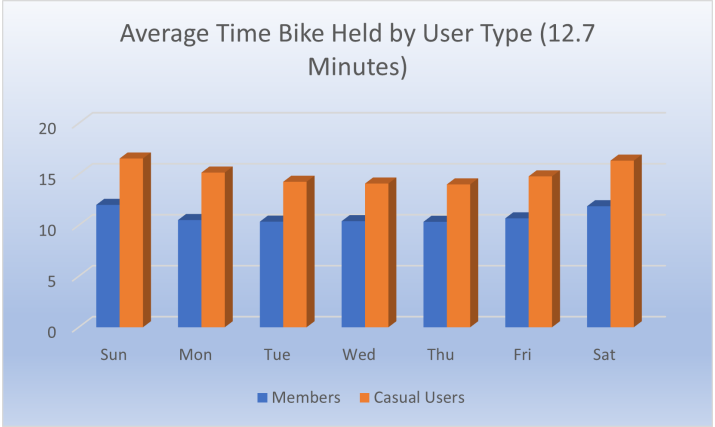
```
knitr::include_graphics('images/Slide_7.png')
```



Average time Bicycles were Used by Month and by Member Type

Average time (mean) Cyclistic Bike-Share Bicycles were Used by Week and by Member Type

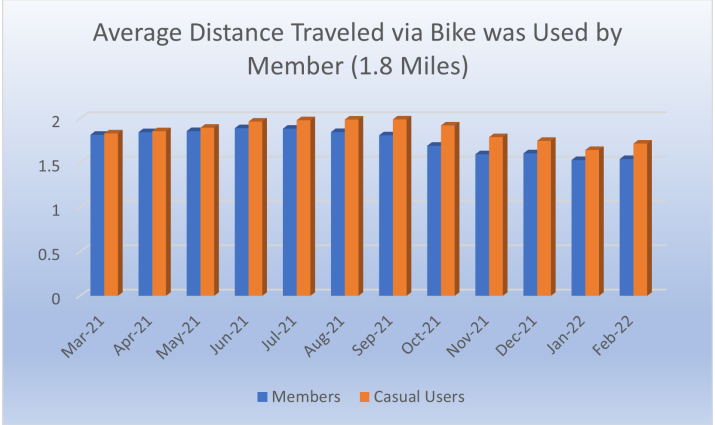
```
knitr::include_graphics('images/Slide_8.png')
```



Average time Bicycles were Used by Week and by Member Type

Average Distance Traveled (mean) Cyclistic Bike-Share Bicycles were Used by Month and by Member Type

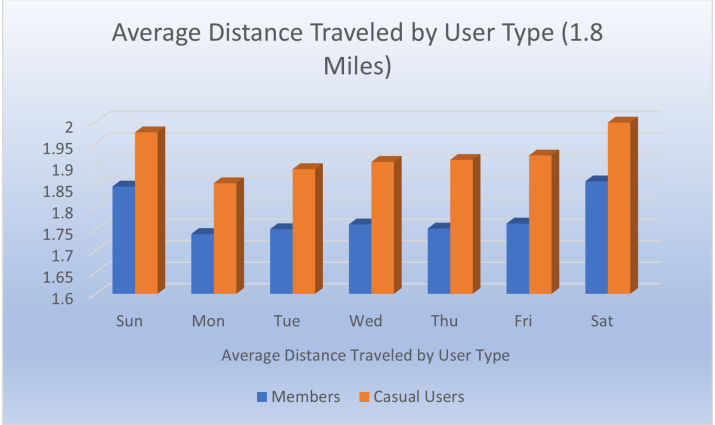
```
knitr::include_graphics('images/Slide_9.png')
```



Average Distance Traveled (mean) Cyclistic Bike-Share Bicycles were Used by Month and by Member Type

Average Distance Traveled (mean) Cyclistic Bike-Share Bicycles were Used by Week and by Member Type

```
knitr::include_graphics('images/Slide_10.png')
```



Average Distance Traveled (mean) Cyclistic Bike-Share Bicycles were Used by Week and by Member Type

Key Takeaways

• Casual users of the bike-share program utilize their bicycles longer than annual members.

• While in possession of their bicycles, casual users travel farther than annual members.

• Casual users are more active on the weekends than annual members, which would suggest that they utilize bicycles for leisure.

• Annual members utilize their bicycles during weekdays, suggesting that their usage is to commute to and from work.

• Annual members utilize their bicycles year-round with a more considerable reduction in January and February.

• Casual users mainly use bike-share bicycles during the warmer months (Apr-Oct).

• Annual members never use docked bicycles and primarily utilize classic and electric bicycles, whereas causal members are likely to use all three types of bicycles – classic, electric, and docked.

Act

Recommendations

• Run promotions for casual members touting the similarities they share with annual members, such as they utilize the bicycles longer and travel longer distances than annual members. It may also be cost-effective for them to become annual members.

• Run promotions addressing those casual users that utilize the bicycles longer, that while saving money as an annual member, they will also be able to use the bicycles on the weekdays for their work commute.

Additional data suggested for further analysis:

For further study, it would be helpful to have data on age and gender, all pricing information, the social-economic status of users, and the marital status of users.

The End.