Study_Case_Cyclistic

2025-01-07

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

###Load work directory Set the working directory in the folder the files were downloaded

```
setwd("/Users/giuliaribeiro/Documents/R_course/Case_Study1")
```

Step 2: Import data

In the chunk below, I will use the `read_csv()` function to import data from one of the .csv in the project folder called "202401-divvy-tripdata.csv" and save it as a data frame called `History_Cyclism_202401`.

```
library(readr)
History_Cyclism_202401 <- read_csv("202401-divvy-tripdata.csv")
```

```
## Rows: 144873 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Step 3: Getting to know your data

First I need to get to know the data and how it is structured. First, I used the `head()` function to preview the columns and the first several rows of data.

```
head(History_Cyclism_202401)
```

```
## # A tibble: 6 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 C1D650626C8C899A electric_bike 2024-01-12 15:30:27 2024-01-12 15:37:59
## 2 EECD38BDB25BFCB0 electric_bike 2024-01-08 15:45:46 2024-01-08 15:52:59
## 3 F4A9CE78061F17F7 electric_bike 2024-01-27 12:27:19 2024-01-27 12:35:19
## 4 0A0D9E15EE50B171 classic_bike  2024-01-29 16:26:17 2024-01-29 16:56:06
## 5 33FFC9805E3EFF9A classic_bike  2024-01-31 05:43:23 2024-01-31 06:09:35
## 6 C96080812CD285C5 classic_bike  2024-01-07 11:21:24 2024-01-07 11:30:03
## # i 9 more variables: start_station_name <chr>, start_station_id <chr>,
## #   end_station_name <chr>, end_station_id <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>
```

In addition to `head()` I can also use the `str()` and `glimpse()` functions to get summaries of each column of the data arranged horizontally.

```
str(History_Cyclism_202401)
```

```
## spc_tbl_ [144,873 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:144873] "C1D650626C8C899A" "EECD38BDB25BFCB0" "F4A9CE78061F17F7" "0A0D
##  $ rideable_type     : chr [1:144873] "electric_bike" "electric_bike" "electric_bike" "classic_bike"
##  $ started_at        : POSIXct[1:144873], format: "2024-01-12 15:30:27" "2024-01-08 15:45:46" ...
##  $ ended_at          : POSIXct[1:144873], format: "2024-01-12 15:37:59" "2024-01-08 15:52:59" ...
##  $ start_station_name: chr [1:144873] "Wells St & Elm St" "Wells St & Elm St" "Wells St & Elm St" "We
##  $ start_station_id  : chr [1:144873] "KA1504000135" "KA1504000135" "KA1504000135" "TA1305000030" ..
##  $ end_station_name  : chr [1:144873] "Kingsbury St & Kinzie St" "Kingsbury St & Kinzie St" "Kingsbu
##  $ end_station_id    : chr [1:144873] "KA1503000043" "KA1503000043" "KA1503000043" "13193" ...
##  $ start_lat         : num [1:144873] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:144873] -87.6 -87.6 -87.6 -87.6 -87.7 ...
##  $ end_lat           : num [1:144873] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:144873] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:144873] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
library(tidyverse)
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.0.2
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
glimpse(History_Cyclism_202401)
```

```
## Rows: 144,873
## Columns: 13
## $ ride_id            <chr> "C1D650626C8C899A", "EECD38BDB25BFCB0", "F4A9CE7806~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2024-01-12 15:30:27, 2024-01-08 15:45:46, 2024-01-~
## $ ended_at           <dttm> 2024-01-12 15:37:59, 2024-01-08 15:52:59, 2024-01-~
## $ start_station_name <chr> "Wells St & Elm St", "Wells St & Elm St", "Wells St~
## $ start_station_id   <chr> "KA1504000135", "KA1504000135", "KA1504000135", "TA~
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Kingsbury St & Kinzie ~
## $ end_station_id     <chr> "KA1503000043", "KA1503000043", "KA1503000043", "13~
## $ start_lat          <dbl> 41.90327, 41.90294, 41.90295, 41.88430, 41.94880, 4~
## $ start_lng          <dbl> -87.63474, -87.63444, -87.63447, -87.63396, -87.675~
## $ end_lat            <dbl> 41.88918, 41.88918, 41.88918, 41.92182, 41.88918, 4~
## $ end_lng            <dbl> -87.63851, -87.63851, -87.63851, -87.64414, -87.638~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

Use colnames() to get the names of the columns in the dataset.

```
colnames(History_Cyclism_202401)
```

```
##  [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"
```

To get more detailed on types and statistics of each variable, run summary

```
summary(History_Cyclism_202401)
```

```
##    ride_id          rideable_type        started_at                    
##  Length:144873      Length:144873      Min.   :2024-01-01 00:00:39.00  
##  Class :character   Class :character   1st Qu.:2024-01-06 19:27:53.00  
##  Mode  :character   Mode  :character   Median :2024-01-13 18:30:35.00  
##                                        Mean   :2024-01-16 07:38:03.93  
##                                        3rd Qu.:2024-01-25 21:03:03.00  
##                                        Max.   :2024-01-31 23:59:40.00  
##                                                                        
##     ended_at                      start_station_name start_station_id  
##  Min.   :2024-01-01 00:04:20.00   Length:144873      Length:144873     
##  1st Qu.:2024-01-06 19:41:11.00   Class :character   Class :character  
##  Median :2024-01-13 18:47:51.00   Mode  :character   Mode  :character  
##  Mean   :2024-01-16 07:53:07.36                                        
##  3rd Qu.:2024-01-25 21:26:12.00                                        
##  Max.   :2024-02-02 00:01:21.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:144873      Length:144873      Min.   :41.65   Min.   :-87.84  
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66  
##  Mode  :character   Mode  :character   Median :41.89   Median :-87.64  
##                                        Mean   :41.90   Mean   :-87.65  
##                                        3rd Qu.:41.93   3rd Qu.:-87.63  
##                                        Max.   :42.07   Max.   :-87.53  
##                                                                        
##     end_lat         end_lng       member_casual     
##  Min.   :41.63   Min.   :-87.86   Length:144873     
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character  
##  Median :41.89   Median :-87.64   Mode  :character  
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.07   Max.   :-87.46
##  NA's   :288     NA's   :288
```

Some packages contain more advanced functions for summarizing and exploring your data. One example is the `skimr` package, which has a number of functions for this purpose. For example, the `skim_without_charts()` function provides a detailed summary of the data. Try running the code below:

```
library(skimr)
skim_without_charts(History_Cyclism_202401)
```

Table 1: Data summary

Name	History_Cyclism_202401
Number of rows	144873
Number of columns	13
Column type frequency:	
character	7
numeric	4
POSIXct	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1.00	16	16	0	144873	0
rideable_type	0	1.00	12	13	0	2	0
start_station_name	19165	0.87	10	64	0	999	0
start_station_id	19165	0.87	3	13	0	988	0
end_station_name	20749	0.86	10	64	0	996	0
end_station_id	20749	0.86	3	35	0	986	0
member_casual	0	1.00	6	6	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
start_lat	0	1	41.90	0.05	41.65	41.88	41.89	41.93	42.07
start_lng	0	1	-87.65	0.03	-87.84	-87.66	-87.64	-87.63	-87.53
end_lat	288	1	41.90	0.05	41.63	41.88	41.89	41.93	42.07
end_lng	288	1	-87.65	0.03	-87.86	-87.66	-87.64	-87.63	-87.46

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2024-01-01 00:00:39	2024-01-31 23:59:40	2024-01-13 18:30:35	137197

skim_variable	n_missing	complete_rate	min	max	median	n_unique
ended_at	0	1	2024-01-01 00:04:20	2024-02-02 00:01:21	2024-01-13 18:47:51	137207

##Checking for NA

alternative method for checking for NA

```
# Count missing values in critical columns
sum(is.na(History_Cyclism_202401$started_at))  # Check for missing values in started_at
```

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

```
sum(is.na(History_Cyclism_202401$ended_at))    # Check for missing values in ended_at
```

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

```
sum(is.na(History_Cyclism_202401$rideable_type))  # Check for missing values in rideable_type
```

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

```
sum(is.na(History_Cyclism_202401$member_casual))  # Check for missing values in member_casual
```

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

```
# Check for missing values across the entire dataset
colSums(is.na(History_Cyclism_202401))
```

```
##            ride_id      rideable_type         started_at           ended_at
##                  0                  0                  0                  0
## start_station_name   start_station_id     end_station_name     end_station_id
##              19165              19165              20749              20749
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0                288                288
##      member_casual
##                  0
```

###Validate data ranges

Ensure that started_at occurs before ended_at and that the dates are within logical bounds.

```
# Check for invalid timestamps
sum(History_Cyclism_202401$ended_at < History_Cyclism_202401$started_at)  # Should return 0 if all rows
```

```
## [1] 20
```

```
# Summarize the date range
range(as.Date(History_Cyclism_202401$started_at))  # Earliest and latest start dates
```

```
## [1] "2024-01-01" "2024-01-31"
```

```
range(as.Date(History_Cyclism_202401$ended_at))    # Earliest and latest end dates
```

```
## [1] "2024-01-01" "2024-02-02"
```

###Latitude and longitude Validate that the latitude and longitude fall within valid ranges:

Latitude: -90 to 90 Longitude: -180 to 180

```
# Check for invalid latitude or longitude
sum(History_Cyclism_202401$start_lat < -90 | History_Cyclism_202401$start_lat > 90)  # Invalid start la
```

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

```r
sum(History_Cyclism_202401$start_lng < -180 | History_Cyclism_202401$start_lng > 180)  # Invalid start
```

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

```r
sum(History_Cyclism_202401$end_lat < -90 | History_Cyclism_202401$end_lat > 90)  # Invalid end latitude
```

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

```r
sum(History_Cyclism_202401$end_lng < -180 | History_Cyclism_202401$end_lng > 180)  # Invalid end longit
```

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

No invalid Latitude, longitude data

##Identify duplicates

```r
# Check for duplicate ride_ids
sum(duplicated(History_Cyclism_202401$ride_id))  # Count duplicates
```

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

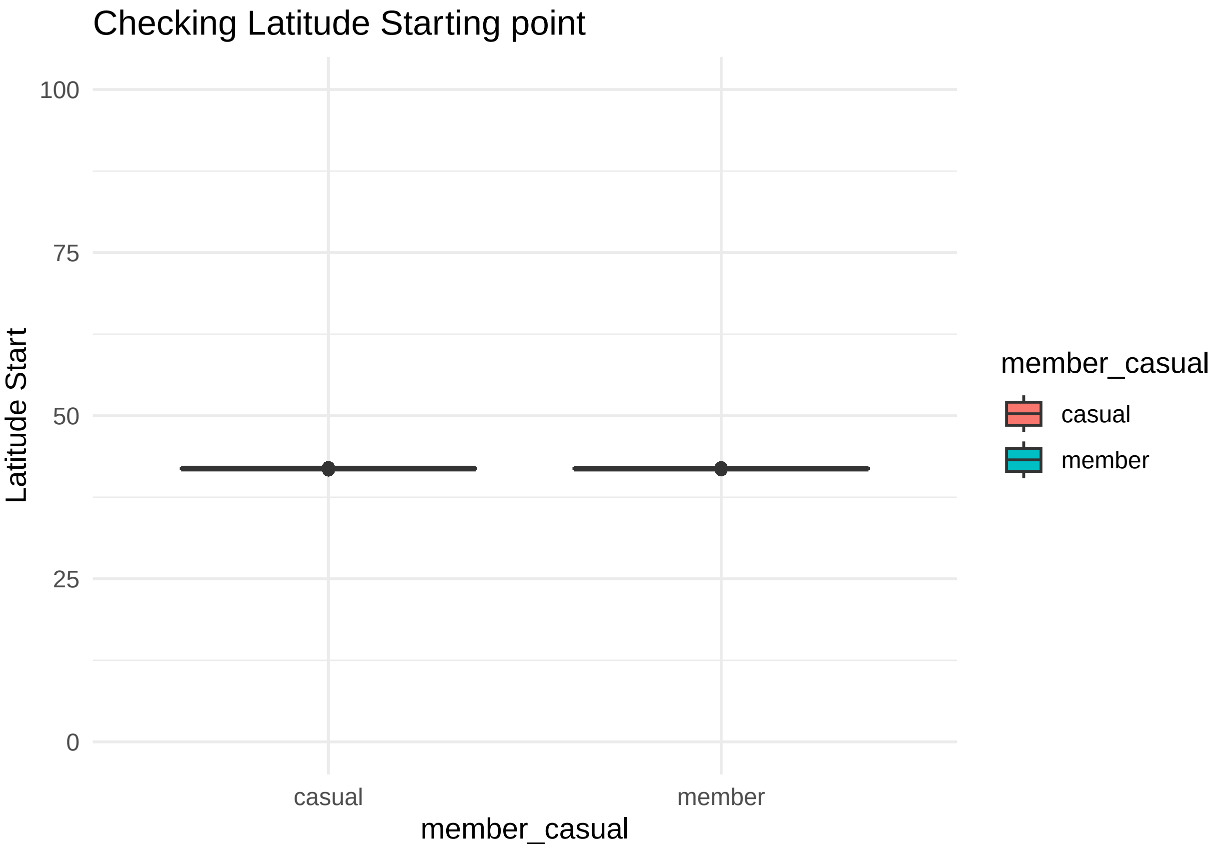
```r
# View duplicate rows if any
History_Cyclism_202401[duplicated(History_Cyclism_202401$ride_id), ]
```

```
## # A tibble: 0 x 13
## # i 13 variables: ride_id <chr>, rideable_type <chr>, started_at <dttm>,
## #   ended_at <dttm>, start_station_name <chr>, start_station_id <chr>,
## #   end_station_name <chr>, end_station_id <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>
```

No duplicates

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

##Handle missing values

The strategy of handle missing values must be analysed with care. Sometimes it is better to remove the entire column because there is a lot of missing values. Other times, just remove the entries with missing data is enough and won't make a lot of difference in the analysis.

```
library(dplyr)
# Remove rows with NA in critical columns like 'started_at', 'ended_at'
#cleaned_data <- raw_data %>% drop_na(started_at, ended_at)
```

##Working with the full dataset

The objective of this work was handling 12 month data. For that, I must first join all datasets that are spread by month and create an extra column for the month

```
# Load required libraries
library(dplyr)
library(readr)
library(lubridate)
```

```
#set working directory
setwd("/Users/giuliaribeiro/Documents/R_course/Case_Study1/")
```

```
# Define the directory where the files are stored
data_dir <- "./monthly_files/"  # Adjust to your folder path
```

```
# Check if all CSV files are in the directory
file_list <- list.files(path = data_dir, pattern = "*.csv", full.names = TRUE)
```

```r
# Read and combine all files
combined_data <- file_list %>%
  lapply(read_csv) %>%  # Read each file into a data frame
  bind_rows()           # Combine all data frames into one
```

```
## Rows: 144873 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 223164 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 301687 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 415025 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 609493 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 710721 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 748962 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 755639 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 821276 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 616281 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 335075 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 178372 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Preview combined data
glimpse(combined_data)
```

```
## Rows: 5,860,568
## Columns: 13
## $ ride_id            <chr> "C1D650626C8C899A", "EECD38BDB25BFCB0", "F4A9CE7806~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2024-01-12 15:30:27, 2024-01-08 15:45:46, 2024-01-~
## $ ended_at           <dttm> 2024-01-12 15:37:59, 2024-01-08 15:52:59, 2024-01-~
## $ start_station_name <chr> "Wells St & Elm St", "Wells St & Elm St", "Wells St~
## $ start_station_id   <chr> "KA1504000135", "KA1504000135", "KA1504000135", "TA~
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Kingsbury St & Kinzie ~
## $ end_station_id     <chr> "KA1503000043", "KA1503000043", "KA1503000043", "13~
## $ start_lat          <dbl> 41.90327, 41.90294, 41.90295, 41.88430, 41.94880, 4~
## $ start_lng          <dbl> -87.63474, -87.63444, -87.63447, -87.63396, -87.675~
## $ end_lat            <dbl> 41.88918, 41.88918, 41.88918, 41.92182, 41.88918, 4~
## $ end_lng            <dbl> -87.63851, -87.63851, -87.63851, -87.64414, -87.638~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
# Ensure the "started_at" column is in datetime format
# If it's already in <dttm> format, this step can be skipped
combined_data <- combined_data %>%
  mutate(started_at = as_datetime(started_at))
```

```
# Extract the month from the datetime column
combined_data <- combined_data %>%
  mutate(month = month(started_at))
```

```
# Preview combined data
glimpse(combined_data)
```

```
## Rows: 5,860,568
## Columns: 14
## $ ride_id            <chr> "C1D650626C8C899A", "EECD38BDB25BFCB0", "F4A9CE7806~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2024-01-12 15:30:27, 2024-01-08 15:45:46, 2024-01-~
## $ ended_at           <dttm> 2024-01-12 15:37:59, 2024-01-08 15:52:59, 2024-01-~
## $ start_station_name <chr> "Wells St & Elm St", "Wells St & Elm St", "Wells St~
## $ start_station_id   <chr> "KA1504000135", "KA1504000135", "KA1504000135", "TA~
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Kingsbury St & Kinzie ~
## $ end_station_id     <chr> "KA1503000043", "KA1503000043", "KA1503000043", "13~
## $ start_lat          <dbl> 41.90327, 41.90294, 41.90295, 41.88430, 41.94880, 4~
## $ start_lng          <dbl> -87.63474, -87.63444, -87.63447, -87.63396, -87.675~
## $ end_lat            <dbl> 41.88918, 41.88918, 41.88918, 41.92182, 41.88918, 4~
## $ end_lng            <dbl> -87.63851, -87.63851, -87.63851, -87.64414, -87.638~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
## $ month              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

```
# Optional: If you want the month as a name instead of a number
combined_data <- combined_data %>%
  mutate(month_name = month(started_at, label = TRUE, abbr = FALSE))
```

```r
# Preview combined data
glimpse(combined_data)
```

```
## Rows: 5,860,568
## Columns: 15
## $ ride_id            <chr> "C1D650626C8C899A", "EECD38BDB25BFCB0", "F4A9CE7806~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2024-01-12 15:30:27, 2024-01-08 15:45:46, 2024-01-~
## $ ended_at           <dttm> 2024-01-12 15:37:59, 2024-01-08 15:52:59, 2024-01-~
## $ start_station_name <chr> "Wells St & Elm St", "Wells St & Elm St", "Wells St~
## $ start_station_id   <chr> "KA1504000135", "KA1504000135", "KA1504000135", "TA~
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Kingsbury St & Kinzie ~
## $ end_station_id     <chr> "KA1503000043", "KA1503000043", "KA1503000043", "13~
## $ start_lat          <dbl> 41.90327, 41.90294, 41.90295, 41.88430, 41.94880, 4~
## $ start_lng          <dbl> -87.63474, -87.63444, -87.63447, -87.63396, -87.675~
## $ end_lat            <dbl> 41.88918, 41.88918, 41.88918, 41.92182, 41.88918, 4~
## $ end_lng            <dbl> -87.63851, -87.63851, -87.63851, -87.64414, -87.638~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
## $ month              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ month_name         <ord> January, January, January, January, January, Januar~
```

```r
# Save combined data as a new file (optional)
write_csv(combined_data, "combined_cyclistic_data.csv")
```

```r
# Inspect summary statistics
summary(combined_data)
```

```
##    ride_id          rideable_type        started_at                    
##  Length:5860568     Length:5860568     Min.   :2024-01-01 00:00:39.00  
##  Class :character   Class :character   1st Qu.:2024-05-20 19:47:53.00  
##  Mode  :character   Mode  :character   Median :2024-07-22 20:36:16.27  
##                                        Mean   :2024-07-17 07:55:47.61  
##                                        3rd Qu.:2024-09-17 20:14:22.56  
##                                        Max.   :2024-12-31 23:56:49.84  
##                                                                        
##     ended_at                       start_station_name start_station_id  
##  Min.   :2024-01-01 00:04:20.00   Length:5860568     Length:5860568    
##  1st Qu.:2024-05-20 20:07:54.75   Class :character   Class :character  
##  Median :2024-07-22 20:53:59.16   Mode  :character   Mode  :character  
##  Mean   :2024-07-17 08:13:06.54                                        
##  3rd Qu.:2024-09-17 20:27:46.02                                        
##  Max.   :2024-12-31 23:59:55.70                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:5860568     Length:5860568     Min.   :41.64   Min.   :-87.91  
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66  
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64  
##                                        Mean   :41.90   Mean   :-87.65  
##                                        3rd Qu.:41.93   3rd Qu.:-87.63  
##                                        Max.   :42.07   Max.   :-87.52  
##                                                                        
##     end_lat          end_lng        member_casual          month       
##  Min.   :16.06   Min.   :-144.05   Length:5860568     Min.   : 1.000  
##  1st Qu.:41.88   1st Qu.: -87.66   Class :character   1st Qu.: 5.000  
##  Median :41.90   Median : -87.64   Mode  :character   Median : 7.000  
##  Mean   :41.90   Mean   : -87.65                     Mean   : 7.019
##  3rd Qu.:41.93   3rd Qu.: -87.63                     3rd Qu.: 9.000
##  Max.   :87.96   Max.   : 152.53                     Max.   :12.000
##  NA's   :7232    NA's   :7232
##      month_name
##  September: 820867
##  August   : 755804
##  July     : 749004
##  June     : 710747
##  October  : 616292
##  May      : 609704
##  (Other)  :1598150
```

```r
library(skimr)
skim_without_charts(combined_data)
```

Table 5: Data summary

Name	combined_data
Number of rows	5860568
Number of columns	15
Column type frequency:	
character	7
factor	1
numeric	5
POSIXct	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1.00	16	16	0	5860357	0
rideable_type	0	1.00	12	16	0	3	0
start_station_name	1073951	0.82	10	64	0	1808	0
start_station_id	1073951	0.82	3	35	0	1763	0
end_station_name	1104653	0.81	10	64	0	1815	0
end_station_id	1104653	0.81	3	35	0	1768	0
member_casual	0	1.00	6	6	0	2	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
month_name	0	1	TRUE	12	Sep: 820867, Aug: 755804, Jul: 749004, Jun: 710747

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
start_lat	0	1	41.90	0.04	41.64	41.88	41.90	41.93	42.07
start_lng	0	1	-87.65	0.03	-87.91	-87.66	-87.64	-87.63	-87.52
end_lat	7232	1	41.90	0.06	16.06	41.88	41.90	41.93	87.96
end_lng	7232	1	-87.65	0.11	-144.05	-87.66	-87.64	-87.63	152.53
month	0	1	7.02	2.67	1.00	5.00	7.00	9.00	12.00

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2024-01-01 00:00:39	2024-12-31 23:56:49	2024-07-22 20:36:16	5649600
ended_at	0	1	2024-01-01 00:04:20	2024-12-31 23:59:55	2024-07-22 20:53:59	5652165

```
# Check for missing values across the entire dataset
colSums(is.na(combined_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
##            1073951            1073951            1104653            1104653
##          start_lat          start_lng            end_lat            end_lng
##                  0                  0               7232               7232
##      member_casual              month         month_name
##                  0                  0                  0
```

```
# Calculate the total number of rows in the dataset
total_rows <- nrow(combined_data)

# Calculate the number of missing values for each column
missing_values <- colSums(is.na(combined_data))

# Calculate the percentage of missing values for each column
missing_percentage <- (missing_values / total_rows) * 100

# Combine the results into a data frame for better readability
missing_summary <- data.frame(
  Column = names(missing_values),
  Missing_Count = missing_values,
  Missing_Percentage = round(missing_percentage, 1) # Rounded to one decimal place
)

# Print the summary
print(missing_summary)
```

```
##                                Column Missing_Count Missing_Percentage
## ride_id                       ride_id             0                0.0
## rideable_type           rideable_type             0                0.0
## started_at                 started_at             0                0.0
## ended_at                     ended_at             0                0.0
## start_station_name start_station_name       1073951               18.3
## start_station_id     start_station_id       1073951              18.3
## end_station_name     end_station_name       1104653              18.8
## end_station_id         end_station_id       1104653              18.8
## start_lat                   start_lat             0               0.0
## start_lng                   start_lng             0               0.0
## end_lat                       end_lat          7232               0.1
## end_lng                       end_lng          7232               0.1
## member_casual           member_casual             0               0.0
## month                           month             0               0.0
## month_name                 month_name             0               0.0
```

```
#Since station names are not the most important data for us and has 18% os missing data, I will replace
combined_data_natreated <- combined_data %>%
  mutate(
    start_station_name = replace_na(start_station_name, "Unknown"),
    start_station_id = replace_na(start_station_id, "Unknown"),
    end_station_name = replace_na(end_station_name, "Unknown"),
    end_station_id = replace_na(end_station_id, "Unknown"),
  )
```

```
# Check for missing values across the entire dataset
colSums(is.na(combined_data_natreated))
```

```
##            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               7232               7232
##      member_casual              month         month_name
##                  0                  0                  0
```

```
# However, about latitude and longitude. Only 0.1% of the values are missing
# In this case I will drop rows with missing values in end_lat and end_lng (minimal data loss).
combined_data_natreated <- combined_data_natreated %>%
  drop_na(end_lat, end_lng)
```

```
# Check for missing values across the entire dataset
colSums(is.na(combined_data_natreated))
```

```
##            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              month         month_name
##                  0                  0                  0
```

```
# Analyze ride length (e.g., by month)
combined_data_natreated <- combined_data_natreated %>%
  mutate(ride_length = as.numeric(difftime(ended_at, started_at, units = "mins")))

monthly_summary <- combined_data_natreated %>%
  group_by(month, member_casual) %>%
  summarize(
    avg_ride_length = mean(ride_length, na.rm = TRUE),
```

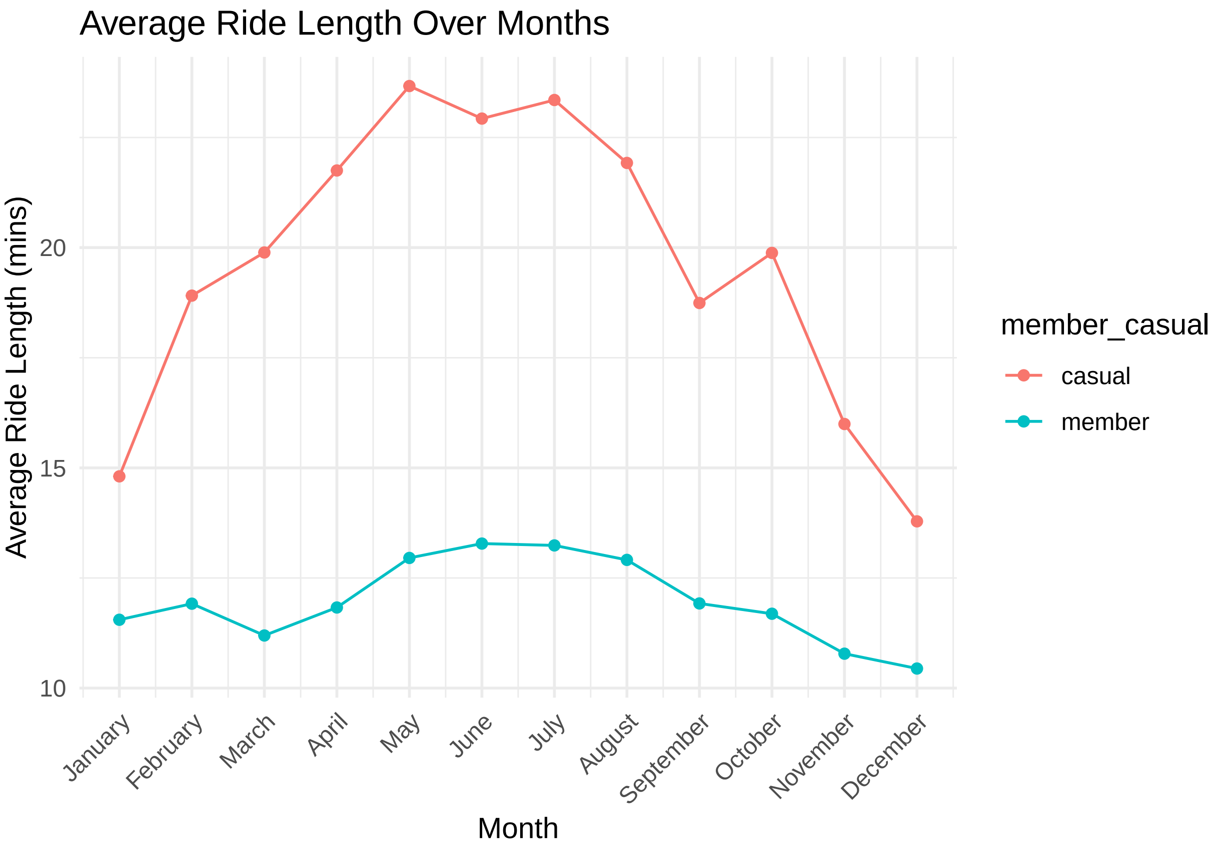
```r
    total_rides = n(),
    .groups = "drop"
  )

# Print summary
print(monthly_summary)
```

```
## # A tibble: 24 x 4
##    month member_casual avg_ride_length total_rides
##    <dbl> <chr>                   <dbl>       <int>
##  1     1 casual                   14.8       24353
##  2     1 member                   11.6      120232
##  3     2 casual                   18.9       46963
##  4     2 member                   11.9      175883
##  5     3 casual                   19.9       82268
##  6     3 member                   11.2      219023
##  7     4 casual                   21.8      131431
##  8     4 member                   11.8      283115
##  9     5 casual                   23.7      230466
## 10     5 member                   13.0      378414
## # i 14 more rows
```

```r
# Visualize ride length over the months
library(ggplot2)

ggplot(monthly_summary, aes(x = month, y = avg_ride_length, color = member_casual)) +
  geom_line() +
  geom_point() +
  labs(title = "Average Ride Length Over Months", x = "Month", y = "Average Ride Length (mins)") +
  scale_x_continuous(breaks = 1:12, labels = month.name) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Average Ride Length Over Months
Average Ride Length (mins)
20
15
10
January
February
March
April
May
June
July
August
September
October
November
December
Month
member_casual
casual
member