Google Capstone Project

Prasad Sawant

2023-01-30

Using R Code to Clean and Analyze Cyclistic Data

Preparing for Cleaning

```
#loading libraries
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.2 ──
## ✓ ggplot2 3.4.0      ✓ purrr   1.0.1
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.5.0
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(chron)
```

```
##
## Attaching package: 'chron'
```

```
## The following objects are masked from 'package:lubridate':
##
##     days, hours, minutes, seconds, years
```

```
library(hms)
```

```
##
## Attaching package: 'hms'
```

```
## The following object is masked from 'package:lubridate':
##
##     hms

library(data.table)

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

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

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

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

#loading cleaned datasets, 12 files from January 2022 - December 2022
jan22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202201.csv")

## Rows: 103770 Columns: 11
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.

feb22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202202.csv")

## Rows: 115609 Columns: 11
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.
```

```
mar22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202203.csv")
```

```
## Rows: 284042 Columns: 11
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.
```

```
apr22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202204.csv")
```

```
## Rows: 371249 Columns: 11
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.
```

```
may22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202205.csv")
```

```
## Rows: 634858 Columns: 11
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.
```

```
jun22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202206.csv")
```

```
## Rows: 769204 Columns: 11
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
```

```
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.

jul22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202207.csv")

## Rows: 823488 Columns: 11
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr  (7): ride_id, rideable_type, started_at, ended_at, member_casual,
start...
## dbl  (1): day_of_week
## time (3): start_time, end_time, ride_length
##
## 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.

aug22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202208.csv")

## Rows: 785932 Columns: 11
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.

sep22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202209.csv")

## Rows: 701339 Columns: 11
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.

oct22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202210.csv")

## Rows: 558685 Columns: 11
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.

nov22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202211.csv")

## Rows: 337735 Columns: 11
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.

dec22 <- read_csv("C:/Users/Owner/OneDrive/Desktop/202212.csv")

## Rows: 181806 Columns: 11
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): ride_id, rideable_type, started_at, ended_at, member_casual
## dbl  (1): day_of_week
## date (2): start_date, end_date
## time (3): start_time, end_time, ride_length
##
## 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.

#merge all dataframes into one year view using rbind
cyclistic_df <-
```

```
rbind(jan22,feb22,mar22,apr22,may22,jun22,jul22,aug22,sep22,oct22,nov22,dec22
)

#view first 6 rows of data set with column headings and data types
head(cyclistic_df)
```

```
## # A tibble: 6 × 11
##   ride_id          ridea…¹ start…² ended…³ membe…⁴ start_date start_…⁵
end_date
##   <chr>            <chr>   <chr>   <chr>   <chr>   <date>     <time>
<date>
## 1 C2F7DD78E82EC8… electr… 13-01-… 13-01-… casual  2022-01-13 11:59:47
2022-01-13
## 2 A6CF8980A652D2… electr… 10-01-… 10-01-… casual  2022-01-10 08:41:56
2022-01-10
## 3 BD0F91DFF741C6… classi… 25-01-… 25-01-… member  2022-01-25 04:53:40
2022-01-25
## 4 CBB80ED4191054… classi… 04-01-… 04-01-… casual  2022-01-04 00:18:04
2022-01-04
## 5 DDC963BFDDA51E… classi… 20-01-… 20-01-… member  2022-01-20 01:31:10
2022-01-20
## 6 A39C6F6CC0586C… classi… 11-01-… 11-01-… member  2022-01-11 18:48:09
2022-01-11
## # … with 3 more variables: end_time <time>, ride_length <time>,
## #   day_of_week <dbl>, and abbreviated variable names ¹rideable_type,
## #   ²started_at, ³ended_at, ⁴member_casual, ⁵start_time
```

Further Data Cleaning

```
#removing any NA's or duplicates, removing invalid data
cyclistic_df <- na.omit(cyclistic_df) #remove rows with NA values
cyclistic_df <- distinct(cyclistic_df) #remove duplicate rows
cyclistic_df <- cyclistic_df[!(cyclistic_df$ride_length <= 0),] #remove rows
where ride length is 0 or negative
```

Adding new columns for Times of Day

```
cyclistic_df$day_of_week <- format(as.Date(cyclistic_df$start_date),"%A")
#modified day_of_week column with the day name

#create a column for different times of the day
h <- hour(cyclistic_df$start_time)
cyclistic_df <- cyclistic_df %>% mutate(time_of_day =
                                          case_when(h == "0" ~ "Night",
                                                    h == "1" ~ "Night",
                                                    h == "2" ~ "Night",
                                                    h == "3" ~ "Night",
                                                    h == "4" ~ "Night",
                                                    h == "5" ~ "Night",
                                                    h == "6" ~ "Morning",
                                                    h == "7" ~ "Morning",
```

```
                                                    h == "8" ~ "Morning",
                                                    h == "9" ~ "Morning",
                                                    h == "10" ~ "Morning",
                                                    h == "11" ~ "Morning",
                                                    h == "12" ~ "Afternoon",
                                                    h == "13" ~ "Afternoon",
                                                    h == "14" ~ "Afternoon",
                                                    h == "15" ~ "Afternoon",
                                                    h == "16" ~ "Afternoon",
                                                    h == "17" ~ "Afternoon",
                                                    h == "18" ~ "Evening",
                                                    h == "19" ~ "Evening",
                                                    h == "20" ~ "Evening",
                                                    h == "21" ~ "Evening",
                                                    h == "22" ~ "Evening",
                                                    h == "23" ~ "Evening")
                                            )
```

Computing Statistics on the Data

```
#calculate ride length and convert to minutes

cyclistic_df$ride_length <-
as.numeric(difftime(cyclistic_df$end_time,cyclistic_df$start_time, units =
"mins"))
cyclistic_df$ride_length <- round(cyclistic_df$ride_length, digits = 1)

#max, min and mean ride times

cyclistic_df <- cyclistic_df[!(cyclistic_df$ride_length <= 0),] #remove rows
where ride length is 0 or negative

cyclistic_df %>%
  group_by(member_casual) %>%
  summarise(max_ride_time = max(ride_length),
            min_ride_time = min(ride_length),
            mean_ride_time = mean(ride_length))

## # A tibble: 2 × 4
##   member_casual max_ride_time min_ride_time mean_ride_time
##   <chr>                <dbl>         <dbl>          <dbl>
## 1 casual               1408.           0.1           21.0
## 2 member               1166.           0.1           12.2

#compare membership types
cyclistic_df %>%
  group_by(member_casual) %>%
  count(member_casual)

## # A tibble: 2 × 2
## # Groups:   member_casual [2]
```

```
##   member_casual       n
##   <chr>           <int>
## 1 casual        2298798
## 2 member        3328810
```

```
#how many people use each type of bike
rideable_share <- cyclistic_df %>%
  group_by(member_casual, rideable_type) %>%
  count(member_casual)
print(rideable_share)
```

```
## # A tibble: 5 × 3
## # Groups:   member_casual, rideable_type [5]
##   member_casual rideable_type       n
##   <chr>         <chr>           <int>
## 1 casual        classic_bike   882723
## 2 casual        docked_bike    173157
## 3 casual        electric_bike 1242918
## 4 member        classic_bike  1701551
## 5 member        electric_bike 1627259
```

```
#total number of rides
nrow(cyclistic_df)
```

```
## [1] 5627608
```

```
#setting up Mode
Mode <- function(x){
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

#find most frequent day of riding
cyclistic_df %>%
group_by(member_casual) %>%
summarise(mode_day_of_week = Mode(day_of_week))
```

```
## # A tibble: 2 × 2
##   member_casual mode_day_of_week
##   <chr>         <chr>
## 1 casual        Saturday
## 2 member        Thursday
```

```
#find the most frequent time of riding
cyclistic_df %>%
group_by(member_casual) %>%
summarise(mode_time = Mode(start_time))
```

```
## # A tibble: 2 × 2
##   member_casual mode_time
##   <chr>         <time>
```

```
## 1 casual        17:05:56
## 2 member        17:04:12
```

Saving the new csv file

```
#download the new data as .csv file
fwrite(cyclistic_df,"C:/Users/Owner/OneDrive/Desktop/Dataset/cyclistic_data.csv")
```