Cyclistic Case Study

Shashank Upadhyay

October 28, 2022

## Attaching Packages

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(lubridate)

##   
## Attaching package: 'lubridate'

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

library(skimr)

library(janitor)

##   
## Attaching package: 'janitor'

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

library(dplyr)

## STEP 1: COLLECTING DATA

Trips\_Sep21 <- read\_csv('202109-divvy-tripdata.csv')  
  
Trips\_Oct21 <- read\_csv('202110-divvy-tripdata.csv')  
  
Trips\_Nov21 <- read\_csv('202111-divvy-tripdata.csv')  
  
Trips\_Dec21 <- read\_csv('202112-divvy-tripdata.csv')  
  
Trips\_Jan22 <- read\_csv('202201-divvy-tripdata.csv')  
  
Trips\_Feb22 <- read\_csv('202202-divvy-tripdata.csv')  
  
Trips\_Mar22 <- read\_csv('202203-divvy-tripdata.csv')  
  
Trips\_Apr22 <- read\_csv('202204-divvy-tripdata.csv')  
  
Trips\_May22 <- read\_csv('202205-divvy-tripdata.csv')  
  
Trips\_Jun22 <- read\_csv('202206-divvy-tripdata.csv')  
  
Trips\_Jul22 <- read\_csv('202207-divvy-tripdata.csv')  
  
Trips\_Aug22 <- read\_csv('202208-divvy-tripdata.csv')

## STEP 2: WRANGLING DATA AND COMBINING INTO A SINGLE FILE

Comparing column names of each of the files

colnames(Trips\_Sep21)  
  
colnames(Trips\_Oct21)  
  
colnames(Trips\_Nov21)  
  
colnames(Trips\_Dec21)  
  
colnames(Trips\_Jan22)  
  
colnames(Trips\_Feb22)  
  
colnames(Trips\_Mar22)  
  
colnames(Trips\_Apr22)  
  
colnames(Trips\_May22)  
  
colnames(Trips\_Jun22)  
  
colnames(Trips\_Jul22)  
  
colnames(Trips\_Aug22)

Inspecting the dataframes and looking for inconguencies

str(Trips\_Sep21)  
  
str(Trips\_Oct21)  
  
str(Trips\_Nov21)  
  
str(Trips\_Dec21)  
  
str(Trips\_Jan22)  
  
str(Trips\_Feb22)  
  
str(Trips\_Mar22)  
  
str(Trips\_Apr22)  
  
str(Trips\_May22)  
  
str(Trips\_Jun22)  
  
str(Trips\_Jul22)  
  
str(Trips\_Aug22)

Comparing column datatype across all dataframes Using compare\_df\_cols

compare\_df\_cols(Trips\_Sep21, Trips\_Oct21, Trips\_Nov21, Trips\_Dec21,  
 Trips\_Jan22, Trips\_Feb22, Trips\_Mar22, Trips\_Apr22, Trips\_May22,  
 Trips\_Jun22, Trips\_Jul22, Trips\_Aug22, return = "mismatch")

Converting end\_station\_id and start\_station\_id to character for easy stackability

Trips\_Sep21 <- mutate(Trips\_Sep21, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))  
Trips\_Nov21 <- mutate(Trips\_Nov21, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))  
Trips\_Oct21 <- mutate(Trips\_Oct21, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))  
Trips\_Dec21 <- mutate(Trips\_Dec21, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))  
Trips\_Jan22 <- mutate(Trips\_Jan22, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))  
Trips\_Feb22 <- mutate(Trips\_Feb22, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))  
  
Trips\_Apr22 <- mutate(Trips\_Apr22, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))  
Trips\_May22 <- mutate(Trips\_May22, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))  
Trips\_Jun22 <- mutate(Trips\_Jun22, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))  
Trips\_Jul22 <- mutate(Trips\_Jul22, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))  
Trips\_Aug22 <- mutate(Trips\_Aug22, end\_station\_id =  
 as.character(end\_station\_id), start\_station\_id =  
 as.character(start\_station\_id))

Double checking column datatype across all data frames

compare\_df\_cols(Trips\_Sep21, Trips\_Oct21, Trips\_Nov21, Trips\_Dec21,  
 Trips\_Jan22, Trips\_Feb22, Trips\_Mar22, Trips\_Apr22, Trips\_May22,  
 Trips\_Jun22, Trips\_Jul22, Trips\_Aug22, return = "mismatch")

Stacking individual data frames into one big data frame

all\_trips <- bind\_rows(Trips\_Sep21, Trips\_Oct21, Trips\_Nov21, Trips\_Dec21,  
 Trips\_Jan22, Trips\_Feb22, Trips\_Mar22, Trips\_Apr22,Trips\_May22,Trips\_Jun22, Trips\_Jul22, Trips\_Aug22)

Removing Unused Columns

all\_trips <- all\_trips %>%  
 select(-c(start\_lat, start\_lng, end\_lat, end\_lng))

Renaming Columns

all\_trips <- all\_trips %>% rename(trip\_id= ride\_id ,ride\_type =  
 rideable\_type  
 ,start\_time = started\_at,end\_time =ended\_at  
 ,from\_station\_name = start\_station\_name  
 ,from\_station\_id = start\_station\_id  
 ,to\_station\_name = end\_station\_name  
 ,to\_station\_id = end\_station\_id  
 ,usertype = member\_casual)

## STEP 3: CLEANING AND ADDING DATA TO PREPARE FOR ANALYSIS

Inspecting the new table that has been created

colnames(all\_trips) #List of column names

dim(all\_trips) #Dimensions of data farme

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

str(all\_trips) #See list of columns and their data types

summary(all\_trips) #Statistical summary of data

skim(all\_trips) #Get summary of data, check missing data

Adding columns that lists the date, month, day, and year of each ride. It will allow us to aggregate ride data for each Month, Day, or Year.

all\_trips$date <- as.Date(all\_trips$start\_time) #Default format is yyyy-mm-dd  
all\_trips$month <- format(as.Date(all\_trips$date), "%m")  
all\_trips$day <- format(as.Date(all\_trips$date), "%d")  
all\_trips$year <- format(as.Date(all\_trips$date), "%Y")  
all\_trips$day\_of\_week <- format(as.Date(all\_trips$date), "%A")

Adding “ride\_length” calculation to all\_trips(in seconds)

all\_trips$ride\_length <- difftime(all\_trips$end\_time,all\_trips$start\_time)

Converting “ride\_length” from Factor to numeric so we can run calculations on the data

is.factor(all\_trips$ride\_length)

all\_trips$ride\_length <- as.numeric(as.character(all\_trips$ride\_length))  
is.numeric(all\_trips$ride\_length)

Removing “Bad/Useless” data

skim(all\_trips$ride\_length)

all\_trips\_v2 <- all\_trips[!(all\_trips$ride\_length<0),]  
skim(all\_trips\_v2)

## STEP 4: CONDUCTING DESCRIPTIVE ANALYSIS

Descriptive analysis on ride\_length

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

Exporting to .CSV file for further analysis

write.csv(all\_trips\_v2, "data.csv")