CapstoneBikers

Me

2023-03-20

# Transformations

Please note that some transformations with power query (excel) were effected, I examined the data set, created columns for duration of ride in mins and used the absolute date function to deal with errors of swapped dates. I appended all the data sets into a single data set and changed the member\_casual list to “member” = Subscriber and “casual”= Customer. I included NA to blank fields in the start\_station and end\_station columns. I removed the start and end stations id as these had errors that slowed down upload and the rendering of the data set. After these, I imported the data set -> harmonizedbikersdata1.xlsx into R. I loaded all the relevant libraries required for the analysis.

# Loading essential packages

library(lubridate)# deals with date functions

##   
## Attaching package: 'lubridate'

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

library(ggplot2)# plots  
library(tidyr)# data cleaning  
library(dplyr)# viewing data

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

library(tinytex)# needed for render  
library(latex2exp)# works with tinytex

# Take a first look at the data

glimpse(harmonizedbikersdata1)

## Rows: 950,691  
## Columns: 12  
## $ ride\_id <chr> "A847FADBBC638E45", "5405B80E996FF60D", "…  
## $ rideable\_type <chr> "docked\_bike", "docked\_bike", "docked\_bik…  
## $ started\_at <dttm> 2020-04-26 17:45:14, 2020-04-17 17:08:54…  
## $ ended\_at <dttm> 2020-04-26 18:12:03, 2020-04-17 17:17:03…  
## $ start\_station\_name <chr> "Eckhart Park", "Drake Ave & Fullerton Av…  
## $ end\_station\_name <chr> "Lincoln Ave & Diversey Pkwy", "Kosciuszk…  
## $ member\_casual <chr> "Subscriber", "Subscriber", "Subscriber",…  
## $ Year <dbl> 2020, 2020, 2020, 2020, 2020, 2020, 2020,…  
## $ `Duratn\_ride(total minutes)` <dbl> 27, 9, 15, 13, 53, 6, 6, 76, 6, 18, 25, 5…  
## $ `Day Name\_start` <chr> "Sunday", "Friday", "Wednesday", "Tuesday…  
## $ `Day Name\_end` <chr> "Sunday", "Friday", "Wednesday", "Tuesday…  
## $ Return\_day <chr> "Same\_day", "Same\_day", "Same\_day", "Same…

# splitting the Combined data and Time column

#The default format is yyyy-mm-dd  
harmonizedbikersdata1$date<- as.Date(harmonizedbikersdata1$started\_at)   
harmonizedbikersdata1$month <- format(as.Date(harmonizedbikersdata1$date), "%m")  
harmonizedbikersdata1$day <- format(as.Date(harmonizedbikersdata1$date), "%d")  
harmonizedbikersdata1$Year <- format(as.Date(harmonizedbikersdata1$date), "%Y")  
harmonizedbikersdata1$`Day Name\_start`<-format(as.Date(harmonizedbikersdata1$date),"%A")

# Another look at the added column names

colnames(harmonizedbikersdata1)

## [1] "ride\_id" "rideable\_type"   
## [3] "started\_at" "ended\_at"   
## [5] "start\_station\_name" "end\_station\_name"   
## [7] "member\_casual" "Year"   
## [9] "Duratn\_ride(total minutes)" "Day Name\_start"   
## [11] "Day Name\_end" "Return\_day"   
## [13] "date" "month"   
## [15] "day"

# Getting rid of the Null values (test data)

# removed the data from both start and end stations which concurrently had No names, created new dataset, it dropped 135,355 entries.  
harmonizedbikersdata2 <- harmonizedbikersdata1[! (harmonizedbikersdata1$start\_station\_name == " NA" | harmonizedbikersdata1$end\_station\_name == "NA" | harmonizedbikersdata1$`Duratn\_ride(total minutes)`<0),]

# Descriptives

summary(harmonizedbikersdata2$`Duratn\_ride(total minutes)`)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 6.0 9.0 15.3 16.0 58721.0

# Components of the summary

aggregate(harmonizedbikersdata2$`Duratn\_ride(total minutes)` ~ harmonizedbikersdata2$member\_casual, FUN = mean)

## harmonizedbikersdata2$member\_casual  
## 1 Customer  
## 2 Subscriber  
## harmonizedbikersdata2$`Duratn\_ride(total minutes)`  
## 1 23.15922  
## 2 12.35488

aggregate(harmonizedbikersdata2$`Duratn\_ride(total minutes)` ~ harmonizedbikersdata2$member\_casual, FUN = median)

## harmonizedbikersdata2$member\_casual  
## 1 Customer  
## 2 Subscriber  
## harmonizedbikersdata2$`Duratn\_ride(total minutes)`  
## 1 11  
## 2 9

aggregate(harmonizedbikersdata2$`Duratn\_ride(total minutes)` ~ harmonizedbikersdata2$member\_casual, FUN = max)

## harmonizedbikersdata2$member\_casual  
## 1 Customer  
## 2 Subscriber  
## harmonizedbikersdata2$`Duratn\_ride(total minutes)`  
## 1 55684  
## 2 58721

aggregate(harmonizedbikersdata2$`Duratn\_ride(total minutes)` ~ harmonizedbikersdata2$member\_casual, FUN = min)

## harmonizedbikersdata2$member\_casual  
## 1 Customer  
## 2 Subscriber  
## harmonizedbikersdata2$`Duratn\_ride(total minutes)`  
## 1 0  
## 2 0

# Compared summary stats by member\_casual

aggregate(harmonizedbikersdata2$`Duratn\_ride(total minutes)` ~ harmonizedbikersdata2$member\_casual + harmonizedbikersdata2$`Duratn\_ride(total minutes)`, FUN = mean)

## harmonizedbikersdata2$member\_casual  
## 1 Customer  
## 2 Subscriber  
## harmonizedbikersdata2$`Duratn\_ride(total minutes)`  
## 1 23.15922  
## 2 12.35488

#What were the good years?

harmonizedbikersdata2 %>% group\_by(member\_casual, Year) %>% summarise(number\_of\_rides = n(),average\_duration = mean(`Duratn\_ride(total minutes)`)) %>% arrange(member\_casual,Year)

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

## # A tibble: 6 × 4  
## # Groups: member\_casual [2]  
## member\_casual Year number\_of\_rides average\_duration  
## <chr> <chr> <int> <dbl>  
## 1 Customer 2020 23588 72.9  
## 2 Customer 2022 162528 17.2  
## 3 Customer 2023 35778 17.6  
## 4 Subscriber 2020 61089 21.9  
## 5 Subscriber 2022 404424 11.4  
## 6 Subscriber 2023 127929 11.0

# What were the most popular ride types?

table(harmonizedbikersdata2$rideable\_type,harmonizedbikersdata2$member\_casual)

##   
## Customer Subscriber  
## classic\_bike 81392 297338  
## docked\_bike 37327 61089  
## electric\_bike 103175 235015

# What does a yearly comparison look like?

table(harmonizedbikersdata2$rideable\_type,harmonizedbikersdata2$member\_casual,harmonizedbikersdata2$Year)

## , , = 2020  
##   
##   
## Customer Subscriber  
## classic\_bike 0 0  
## docked\_bike 23588 61089  
## electric\_bike 0 0  
##   
## , , = 2022  
##   
##   
## Customer Subscriber  
## classic\_bike 65908 223008  
## docked\_bike 11588 0  
## electric\_bike 85032 181416  
##   
## , , = 2023  
##   
##   
## Customer Subscriber  
## classic\_bike 15484 74330  
## docked\_bike 2151 0  
## electric\_bike 18143 53599

# Analyze rides by member\_casual and weekday

#analyzed ridership data by type and weekday  
harmonizedbikersdata2 %>% mutate(`Day Name\_start`= wday(started\_at, label = TRUE)) %>% group\_by(member\_casual, `Day Name\_start`) %>%  
summarise(number\_of\_rides = n(),average\_duration = mean(`Duratn\_ride(total minutes)`)) %>% arrange(member\_casual, `Day Name\_start`)

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

## # A tibble: 14 × 4  
## # Groups: member\_casual [2]  
## member\_casual `Day Name\_start` number\_of\_rides average\_duration  
## <chr> <ord> <int> <dbl>  
## 1 Customer Sun 35524 27.9  
## 2 Customer Mon 24690 20.3  
## 3 Customer Tue 34371 22.5  
## 4 Customer Wed 34015 18.1  
## 5 Customer Thu 34921 21.7  
## 6 Customer Fri 29339 25.1  
## 7 Customer Sat 29034 26.3  
## 8 Subscriber Sun 64680 15.0  
## 9 Subscriber Mon 85344 11.6  
## 10 Subscriber Tue 113454 11.8  
## 11 Subscriber Wed 106794 11.4  
## 12 Subscriber Thu 92118 12.4  
## 13 Subscriber Fri 73831 11.9  
## 14 Subscriber Sat 57221 13.9

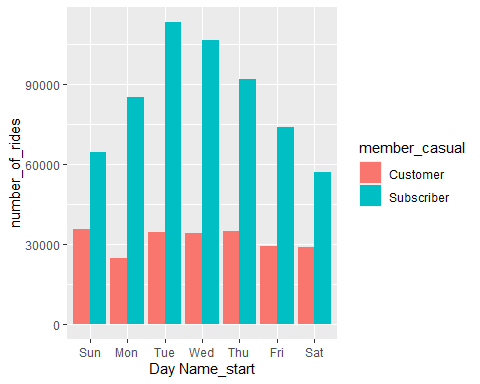
# Rearrange sequence

# reordering the output of weekdays above to a sequence  
harmonizedbikersdata2$`Day Name\_start` <- ordered(harmonizedbikersdata2$`Day Name\_start`, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))

# Rides by weekday

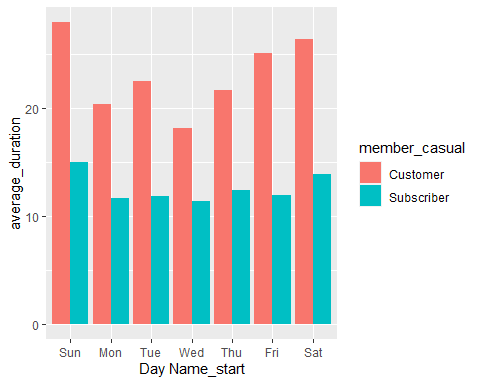
harmonizedbikersdata2 %>% mutate(`Day Name\_start` = wday(started\_at, label = TRUE)) %>% group\_by(member\_casual, `Day Name\_start`) %>% summarise(number\_of\_rides = n(),average\_duration = mean(`Duratn\_ride(total minutes)`)) %>% arrange(member\_casual, `Day Name\_start`) %>% ggplot(aes(x = `Day Name\_start`, y = number\_of\_rides, fill = member\_casual)) + geom\_col(position = "dodge")

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

 # Duration of rides by member\_casual and weekday

#Visualized number of rides by average\_duration  
harmonizedbikersdata2 %>% mutate(`Day Name\_start` = wday(started\_at, label = TRUE)) %>% group\_by(member\_casual, `Day Name\_start`) %>% summarise(number\_of\_rides = n(),average\_duration = mean(`Duratn\_ride(total minutes)`)) %>% arrange(member\_casual) %>% ggplot(aes(x= `Day Name\_start`,y= average\_duration, fill =member\_casual)) + geom\_col(position = "dodge")

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



##Findings #The average bike ride was 15 minutes and the longest time a bike was kept was 41 days. Customers on average spent twice the time( approx. 24 mins) on rides than subscribers-( 12 minutes). However, the median time for both were close at 11 mins for customers and 9 mins for subscribers. Sundays were the top days for long bike rides. A trend shows that bikes are been ridden less by year, this could however be due to the data used , which was picking the latest date in a year for analysis. Classic bikes were the most popular choice (46%), followed by electric bikes(41%) and least by docked bikes(12%).