Case Study 1

Rithvik

2023-02-15

if (!require("tidyverse")) install.packages("tidyverse")

## Loading required package: tidyverse

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.3.0 ✔ 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(tidyverse)  
if (!require("lubridate")) install.packages("lubridate")

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

library(lubridate)  
if (!require("geosphere")) install.packages("geosphere")

## Loading required package: geosphere

library(geosphere)  
if (!require("sf")) install.packages("sf")

## Loading required package: sf  
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf\_use\_s2() is TRUE

library(sf)  
if (!require("mapview")) install.packages("mapview")

## Loading required package: mapview

library(mapview)

# Ask

# Prepare

# Process

tripdata <-  
 list.files(path = "./", pattern = "2022.\*divvy.\*.csv") %>%   
 map\_df(~read\_csv(.))

## Rows: 103770 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: 115609 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: 284042 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: 371249 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: 634858 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: 769204 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: 823488 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: 785932 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: 701339 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: 558685 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: 337735 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: 181806 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.

Lets look at the data

names(tripdata)

## [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"

str(tripdata)

## spc\_tbl\_ [5,667,717 × 13] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ride\_id : chr [1:5667717] "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED419105406" ...  
## $ rideable\_type : chr [1:5667717] "electric\_bike" "electric\_bike" "classic\_bike" "classic\_bike" ...  
## $ started\_at : POSIXct[1:5667717], format: "2022-01-13 11:59:47" "2022-01-10 08:41:56" ...  
## $ ended\_at : POSIXct[1:5667717], format: "2022-01-13 12:02:44" "2022-01-10 08:46:17" ...  
## $ start\_station\_name: chr [1:5667717] "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & Fullerton Ave" "Clark St & Bryn Mawr Ave" ...  
## $ start\_station\_id : chr [1:5667717] "525" "525" "TA1306000016" "KA1504000151" ...  
## $ end\_station\_name : chr [1:5667717] "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave & Fullerton Ave" "Paulina St & Montrose Ave" ...  
## $ end\_station\_id : chr [1:5667717] "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...  
## $ start\_lat : num [1:5667717] 42 42 41.9 42 41.9 ...  
## $ start\_lng : num [1:5667717] -87.7 -87.7 -87.7 -87.7 -87.6 ...  
## $ end\_lat : num [1:5667717] 42 42 41.9 42 41.9 ...  
## $ end\_lng : num [1:5667717] -87.7 -87.7 -87.7 -87.7 -87.6 ...  
## $ member\_casual : chr [1:5667717] "casual" "casual" "member" "casual" ...  
## - 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>

We need to know 1. Duration of the ride - duration 2. Date, extract from timestamp ‘started\_at’ We will ignore corner cases where rides were taken around midnight, and use only started\_at for the same - date 2. Weekday - Sunday, Monday etc. that the ride was taken. day\_of\_the\_week 3. Whether the ride was a weekday ride(Mon..Fri) or a weekend ride(Sat,Sun).

For duration, it is simple. Luckily, the started\_at and ended\_at are both in POSIX\_ct format as timestamps.

tripdata$duration <- tripdata$ended\_at - tripdata$started\_at  
tripdata <- tripdata[tripdata$duration>0,]

Date can easily be extracted, and the

tripdata$date <- lubridate::date(tripdata$started\_at)

From this, we get the day\_of\_the\_week

tripdata$day\_of\_the\_week <- as.factor(wday(tripdata$started\_at, label=TRUE))

We now need to decide whether the day is a weekday or weekend. This is simple:

weekday\_or\_end <- function(day)   
 ifelse(day=="Sat" | day=="Sun","Weekend","Weekday")

We define the simple one-line function. Then we plug it into the assignment below. Note that these values should be factors

tripdata$weekday\_or\_end <- as.factor(weekday\_or\_end(as.character(tripdata$day\_of\_the\_week)))

Other columns we should have as factors are: rideable\_type and member\_casual

tripdata$rideable\_type <- as.factor(tripdata$rideable\_type)  
tripdata$member\_casual <- as.factor(tripdata$member\_casual)

str(tripdata)

## tibble [5,667,186 × 17] (S3: tbl\_df/tbl/data.frame)  
## $ ride\_id : chr [1:5667186] "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED419105406" ...  
## $ rideable\_type : Factor w/ 3 levels "classic\_bike",..: 3 3 1 1 1 1 1 1 3 1 ...  
## $ started\_at : POSIXct[1:5667186], format: "2022-01-13 11:59:47" "2022-01-10 08:41:56" ...  
## $ ended\_at : POSIXct[1:5667186], format: "2022-01-13 12:02:44" "2022-01-10 08:46:17" ...  
## $ start\_station\_name: chr [1:5667186] "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & Fullerton Ave" "Clark St & Bryn Mawr Ave" ...  
## $ start\_station\_id : chr [1:5667186] "525" "525" "TA1306000016" "KA1504000151" ...  
## $ end\_station\_name : chr [1:5667186] "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave & Fullerton Ave" "Paulina St & Montrose Ave" ...  
## $ end\_station\_id : chr [1:5667186] "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...  
## $ start\_lat : num [1:5667186] 42 42 41.9 42 41.9 ...  
## $ start\_lng : num [1:5667186] -87.7 -87.7 -87.7 -87.7 -87.6 ...  
## $ end\_lat : num [1:5667186] 42 42 41.9 42 41.9 ...  
## $ end\_lng : num [1:5667186] -87.7 -87.7 -87.7 -87.7 -87.6 ...  
## $ member\_casual : Factor w/ 2 levels "casual","member": 1 1 2 1 2 2 2 2 2 2 ...  
## $ duration : 'difftime' num [1:5667186] 177 261 261 896 ...  
## ..- attr(\*, "units")= chr "secs"  
## $ date : Date[1:5667186], format: "2022-01-13" "2022-01-10" ...  
## $ day\_of\_the\_week : Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<..: 5 2 3 3 5 3 1 7 2 6 ...  
## $ weekday\_or\_end : Factor w/ 2 levels "Weekday","Weekend": 1 1 1 1 1 1 2 2 1 1 ...

# Analyse

weekly\_trips <- tripdata %>%  
 group\_by(weekday\_or\_end, member\_casual,date) %>%  
 summarise(sum = sum(duration), mean = mean(duration), mode = mode(duration), max = max(duration), min = min(duration), count = sum(duration>10,na.rm=TRUE))

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

bike\_prefs <- tripdata %>%  
 group\_by(rideable\_type, member\_casual,date) %>%  
 summarise(sum = sum(duration), mean = mean(duration), mode = mode(duration), max = max(duration), min = min(duration),count = sum(duration>10,na.rm=TRUE))

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

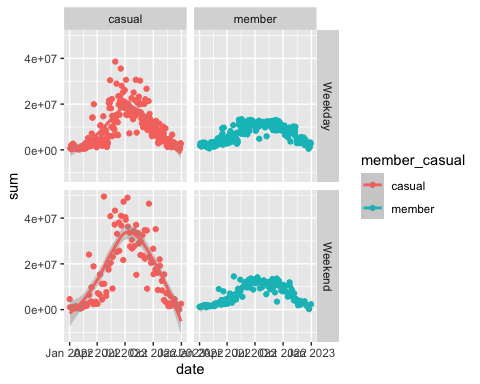
trips\_by\_weekday <- tripdata %>%  
 group\_by(day\_of\_the\_week, member\_casual,date) %>%  
 summarise(sum = sum(duration), mean = mean(duration), mode = mode(duration), max = max(duration), min = min(duration), count = sum(duration>10,na.rm=TRUE))

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

## Constructing plots

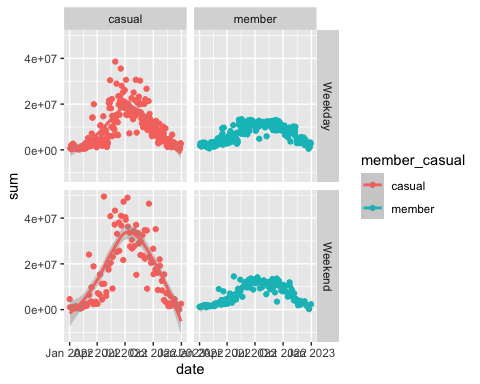
gg\_weekly\_sums <- ggplot(data = weekly\_trips, aes(x=date, y=sum, col=member\_casual)) +  
 facet\_grid(weekday\_or\_end~member\_casual) +  
 geom\_point() +  
 geom\_smooth()  
print(gg\_weekly\_sums)

## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



gg\_weekly\_means <- ggplot(data = weekly\_trips, aes(x=date, y=sum, col=member\_casual)) +  
 facet\_grid(weekday\_or\_end~member\_casual) +  
 geom\_point() +  
 geom\_smooth()  
print(gg\_weekly\_means)

## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



g <- ggplot(data = weekly\_trips, aes(x=date, y=mean, col=member\_casual)) +  
 facet\_grid(weekday\_or\_end~member\_casual) +  
 geom\_point() +  
 geom\_smooth()  
print(g)

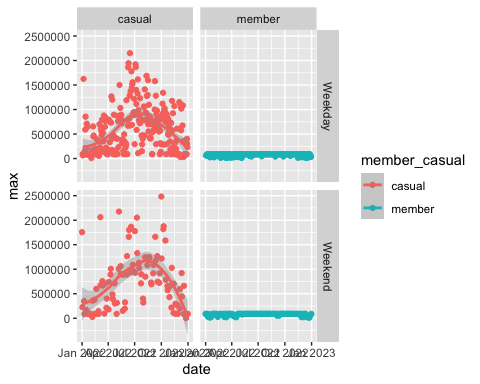
## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



Drastic difference:-

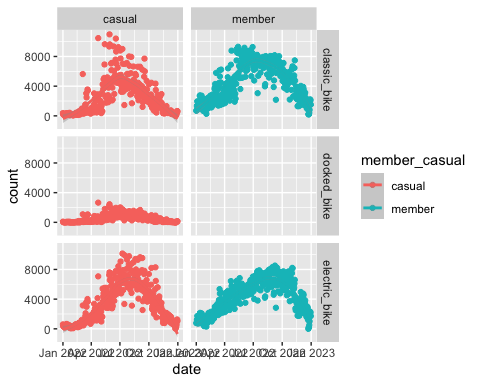
gg\_week\_max <- ggplot(data = weekly\_trips, aes(x=date, y=max, col=member\_casual)) +  
 facet\_grid(weekday\_or\_end~member\_casual) +  
 geom\_point() +  
 geom\_smooth()  
print(gg\_week\_max)

## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



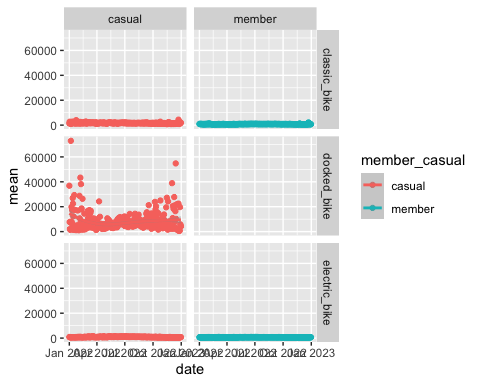
gg\_bike\_counts <- ggplot(data = bike\_prefs, aes(x=date, y=count, col=member\_casual)) +  
 facet\_grid(rideable\_type~member\_casual) +  
 geom\_point() +  
 geom\_smooth()  
print(gg\_bike\_counts)

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



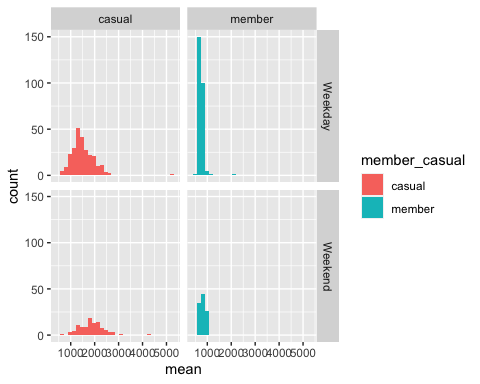
gg\_bike\_mean <- ggplot(data = bike\_prefs, aes(x=date, y=mean, col=member\_casual)) +  
 facet\_grid(rideable\_type~member\_casual) +  
 geom\_point() +  
 geom\_smooth()  
print(gg\_bike\_mean)

## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



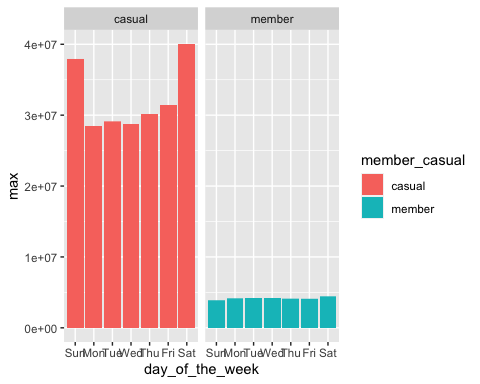
gg\_weekday\_hist <- ggplot(data = weekly\_trips, aes(x=mean, fill=member\_casual)) +  
 facet\_grid(weekday\_or\_end~member\_casual) +  
 geom\_histogram()  
print(gg\_weekday\_hist)

## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



gg\_weekday\_bar <- ggplot(data = trips\_by\_weekday, aes(x=day\_of\_the\_week, y=max, fill=member\_casual)) +  
 facet\_grid(.~member\_casual) +  
 geom\_col()  
print(gg\_weekday\_bar)

## Don't know how to automatically pick scale for object of type <difftime>.  
## Defaulting to continuous.



sum\_trips\_by\_weekday <- trips\_by\_weekday[,c("day\_of\_the\_week","sum")]