Google-DA Project1

Gandikota Sreedhar

January 2023

#====================== # STEP 1: COLLECT DATA #======================

Installing packages

```
#install.packages("tidyverse")
#install.packages("markdown")
#install.packages("sqldf")
#install.packages("maps")
#install.packages("rgdal")
#install.packages("ggrepel")
library("tidyverse")
```

```
## Warning: package ' tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages --------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package ' tibble' was built under R version 3.6.3
```

```
## Warning: package ' tidyr' was built under R version 3.6.3
```

```
## Warning: package ' readr' was built under R version 3.6.3
```

```
## Warning: package ' purrr' was built under R version 3.6.3
```

```
## Warning: package ' dplyr' was built under R version 3.6.3
```

```
## Warning: package ' forcats' was built under R version 3.6.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("lubridate")
```

```
## Warning: package ' lubridate' was built under R version 3.6.3
```

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

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

```
library("markdown")
```

```
## Warning: package ' markdown' was built under R version 3.6.3
```

```
library("sqldf")
```

```
## Warning: package ' sqldf' was built under R version 3.6.3

## Loading required package: gsubfn

## Warning: package ' gsubfn' was built under R version 3.6.3

## Loading required package: proto

## Warning: package ' proto' was built under R version 3.6.3

## Loading required package: RSQLite

## Warning: package ' RSQLite' was built under R version 3.6.3
```

```
library("maps")
```

```
## Warning: package ' maps' was built under R version 3.6.3

##
## Attaching package: ' maps'

## The following object is masked from ' package:purrr' :
##
##     map
```

```
library("rgdal")
```

```
## Warning: package ' rgdal' was built under R version 3.6.3

## Loading required package: sp

## Warning: package ' sp' was built under R version 3.6.3
```

```
## rgdal: version: 1.5-23, (SVN revision 1121)
## Geospatial Data Abstraction Library extensions to R successfully loaded##
Loaded GDAL runtime: GDAL 3.2.1, released 2020/12/29
## Path to GDAL shared files: C:/Users/TRICK/anaconda3/envs/rstudio/lib/R/library/rgdal/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 7.2.1, January 1st, 2021, [PJ_VERSION: 721]
## Path to PROJ shared files: C:/Users/TRICK/anaconda3/envs/rstudio/lib/R/library/rgdal/proj
## PROJ CDN enabled: FALSE
## Linking to sp version:1.4-5
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading rgdal.
## Overwritten PROJ_LIB was C:/Users/TRICK/anaconda3/envs/rstudio/lib/R/library/rgdal/proj
```

```
library("ggrepel")
```

```
## Warning: package ' ggrepel' was built under R version 3.6.3
```

Setting working directory, and creating dataframes for each .csv file.

```
JAN2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
FEB2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
MAR2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
APR2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
MAY2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
JUNE2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/C SV DATA OF P
JULY2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/C SV DATA OF P
AUG2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
SEP2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
OCT2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
NOV2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
DEC2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
JAN2022 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CS V DATA OF PA
```

Glimpsing a dataframe, to see if data types from excel were preserved (they weren't)

```
glimpse(JAN2021)
```

```
## Rows: 96,828
## Columns: 15
## $ ride_id            <fct> A3F8D895163BBB49, 0D139A3203274B87, C7AE8E9CDB~
## $ rideable_type      <fct> electric_bike, classic_bike, classic_bike, ele~
## $ started_at         <fct> 01-01-2021 00:02, 01-01-2021 00:02, 01-01-2021~
## $ ended_at           <fct> 01-01-2021 00:12, 01-01-2021 00:08, 01-01-2021~
## $ start_station_name <fct> , State St & 33rd St, Lakeview Ave & Fullerton~
## $ start_station_id <fct> , 13216, TA1309000019, 13085, TA1308000012, TA~
## $ end_station_name  <fct> , MLK Jr Dr & 29th St, Ritchie Ct & Banks St, ~
## $ end_station_id     <fct> , TA1307000139, KA1504000134, , TA1308000012, ~
```

```
## $ start_lat          <dbl> 41.98000, 41.83473, 41.92586, 41.92953, 41.963~
## $ start_lng          <dbl> -87.65000, -87.62581, -87.63897, -87.70790, -8~
## $ end_lat            <dbl> 41.98000, 41.84205, 41.90687, 41.92000, 41.963~
## $ end_lng            <dbl> -87.66000, -87.61700, -87.62622, -87.72000, -8~
## $ member_casual      <fct> member, member, member, member, member, casual~
## $ ride_length        <fct> 00:10:34, 00:06:15, 00:19:41, 00:07:53, 00:00:~
## $ day_of_week        <int> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6~
```

Merging all the dataframes together

```
tot_rows <- nrow(JAN2021)+nrow(FEB2021)+nrow(MAR2021)+nrow(APR2021)+nrow(MAY2021)+nrow(JUNE2021)+nrow(J
```

```
#=====================================================  # STEP 2:
WRANGLE DATA AND COMBINE INTO A SINGLE FILE #==================================
```

CREATING THE BIND

```
df <- do.call("rbind", list(JAN2021,FEB2021,MAR2021,APR2021,MAY2021,JUNE2021,JULY2021,AUG2021,SEP2021,O
```

Checking if the rows matches with merged dataframe or not

```
if (tot_rows == nrow(df)){
  print("Binding was sucessfull, data verified.")
} else{
  print("Error, please verify your data.")
}
```

```
## [1] "Binding was sucessfull, data verified."
```

UseING sqldf to collect information and store it in a dataframe

creating two dataframes with top 5 start & end stations

####Top 5 starting stations for members

```
mem_start_geo <- sqldf("SELECT member_casual, start_station_name AS Start,
                start_lat AS Starting_Latitude,
                start_lng As Starting_Longitude, count(start_station_name) AS Num_Trips
                FROM df
                WHERE start_station_name IS NOT ''
                AND member_casual = 'member'
                GROUP BY start_station_name
                ORDER BY count(start_station_name) DESC
                LIMIT 5", method = "auto")
mem_start_geo
```

```
##   member_casual                    Start Starting_Latitude
## 1        member        Clark St & Elm St          41.90278
## 2        member Kingsbury St & Kinzie St          41.88918
## 3        member    Wells St & Concord Ln          41.91213
## 4        member        Wells St & Elm St          41.90322
## 5        member    Dearborn St & Erie St          41.89410
##   Starting_Longitude Num_Trips
## 1          -87.63161     25454
## 2          -87.63851     24538
## 3          -87.63466     24242
## 4          -87.63432     21538
## 5          -87.62922     20102
```

####Top 5 starting stations for casuals

```
cas_start_geo <- sqldf("SELECT member_casual, start_station_name AS Start,
                start_lat AS Starting_Latitude, start_lng As Starting_Longitude,
                count(start_station_name) AS Num_Trips
                FROM df
                WHERE start_station_name IS NOT ''
                AND member_casual = 'casual'
                GROUP BY start_station_name
                ORDER BY count(start_station_name) DESC
                LIMIT 5", method = "auto")
cas_start_geo
```

```
##   member_casual                   Start Starting_Latitude
## 1        casual Streeter Dr & Grand Ave          41.89228
## 2        casual         Millennium Park          41.88103
## 3        casual   Michigan Ave & Oak St          41.90096
## 4        casual          Shedd Aquarium          41.86723
## 5        casual     Theater on the Lake          41.92628
##   Starting_Longitude Num_Trips
## 1          -87.61204     66474
## 2          -87.62408     33668
## 3          -87.62378     29812
## 4          -87.61535     23340
## 5          -87.63083     21369
```

###Binding the two tables into a dataframe, and viewing it

```
start_geo <- rbind(mem_start_geo, cas_start_geo)
start_geo
```

```
##    member_casual                    Start Starting_Latitude
## 1         member        Clark St & Elm St          41.90278
## 2         member Kingsbury St & Kinzie St          41.88918
## 3         member    Wells St & Concord Ln          41.91213
## 4         member        Wells St & Elm St          41.90322
## 5         member    Dearborn St & Erie St          41.89410
## 6         casual  Streeter Dr & Grand Ave          41.89228
## 7         casual          Millennium Park          41.88103
```

```
## 8         casual    Michigan Ave & Oak St          41.90096
## 9         casual           Shedd Aquarium          41.86723
## 10        casual      Theater on the Lake          41.92628
##    Starting_Longitude Num_Trips
## 1           -87.63161     25454
## 2           -87.63851     24538
## 3           -87.63466     24242
## 4           -87.63432     21538
## 5           -87.62922     20102
## 6           -87.61204     66474
## 7           -87.62408     33668
## 8           -87.62378     29812
## 9           -87.61535     23340
## 10          -87.63083     21369
```

#========== # STEP 3: CLEAN UP AND ADD DATA TO PREPARE FOR ANALYSIS #==========

Changing the datatype of the coordinates to real numbers to use for plots

```
start_geo$Starting_Latitude = as.numeric(gsub(",",".",start_geo$Starting_Latitude,fixed=TRUE))
start_geo$Starting_Longitude = as.numeric(gsub(",",".",start_geo$Starting_Longitude,fixed=TRUE))
```

####Top 5 ending stations for members

```
mem_end_geo <- sqldf("SELECT member_casual, end_station_name AS End,
                end_lat AS Ending_Latitude,
                end_lng As Ending_Longitude, count(end_station_name) AS Num_Trips
                FROM df
                WHERE end_station_name IS NOT ''
                AND member_casual = 'member'
                GROUP BY end_station_name
                ORDER BY count(end_station_name) DESC
                LIMIT 5", method = "auto")
mem_end_geo
```

```
##   member_casual                     End Ending_Latitude Ending_Longitude
## 1        member        Clark St & Elm St        41.90297        -87.63128
## 2        member    Wells St & Concord Ln        41.91212        -87.63485
## 3        member Kingsbury St & Kinzie St        41.88947        -87.63850
## 4        member        Wells St & Elm St        41.90322        -87.63432
## 5        member    Dearborn St & Erie St        41.89414        -87.62879
##   Num_Trips
## 1     25592
## 2     24956
## 3     24525
## 4     22167
## 5     20838
```

####Top 5 ending stations for casuals

```
cas_end_geo <- sqldf("SELECT member_casual, end_station_name AS End,
                end_lat AS Ending_Latitude, end_lng As Ending_Longitude,
                count(end_station_name) AS Num_Trips
                FROM df
                WHERE end_station_name IS NOT ''
                AND member_casual = 'casual'
                GROUP BY end_station_name
                ORDER BY count(end_station_name) DESC
                LIMIT 5", method = "auto")
cas_end_geo
```

```
##   member_casual                    End Ending_Latitude Ending_Longitude
## 1        casual Streeter Dr & Grand Ave        41.89228        -87.61204
## 2        casual         Millennium Park        41.88103        -87.62408
## 3        casual   Michigan Ave & Oak St        41.90096        -87.62378
## 4        casual     Theater on the Lake        41.92628        -87.63097
## 5        casual          Shedd Aquarium        41.86723        -87.61535
##   Num_Trips
## 1     68789
## 2     34683
## 3     31242
## 4     22771
## 5     21648
```

###Binding the two tables into a dataframe, and viewing it

```
end_geo <- rbind(mem_end_geo, cas_end_geo)
end_geo
```

```
##    member_casual                      End Ending_Latitude Ending_Longitude
## 1         member        Clark St & Elm St        41.90297        -87.63128
## 2         member    Wells St & Concord Ln        41.91212        -87.63485
## 3         member Kingsbury St & Kinzie St        41.88947        -87.63850
## 4         member        Wells St & Elm St        41.90322        -87.63432
## 5         member    Dearborn St & Erie St        41.89414        -87.62879
## 6         casual  Streeter Dr & Grand Ave        41.89228        -87.61204
## 7         casual          Millennium Park        41.88103        -87.62408
## 8         casual    Michigan Ave & Oak St        41.90096        -87.62378
## 9         casual      Theater on the Lake        41.92628        -87.63097
## 10        casual           Shedd Aquarium        41.86723        -87.61535
##    Num_Trips
## 1      25592
## 2      24956
## 3      24525
## 4      22167
## 5      20838
## 6      68789
## 7      34683
## 8      31242
## 9      22771
## 10     21648
```

Changing the datatype of the coordinates to real numbers to use for plots

```
end_geo$Ending_Latitude = as.numeric(gsub(",",".",end_geo$Ending_Latitude, fixed=TRUE))
end_geo$Ending_Longitude = as.numeric(gsub(",",".",end_geo$Ending_Longitude, fixed=TRUE))
```

#==================================== # STEP 4: CONDUCT DESCRIPTIVE ANALYSIS #====================================

Creating a geolocation map of the top 5 start and end stations

###Getting a shapefile of Chicago, and fortifying it into a dataframe

```
chicago_map <- readOGR(dsn="C:/Users/TRICK/Downloads/Boundaries - Community Areas (current)", layer="ge
```

```
## Warning in OGRSpatialRef(dsn, layer, morphFromESRI = morphFromESRI, dumpSRS
## = dumpSRS, : Discarded datum WGS84 in Proj4 definition: +proj=longlat
## +ellps=WGS84 +no_defs
```

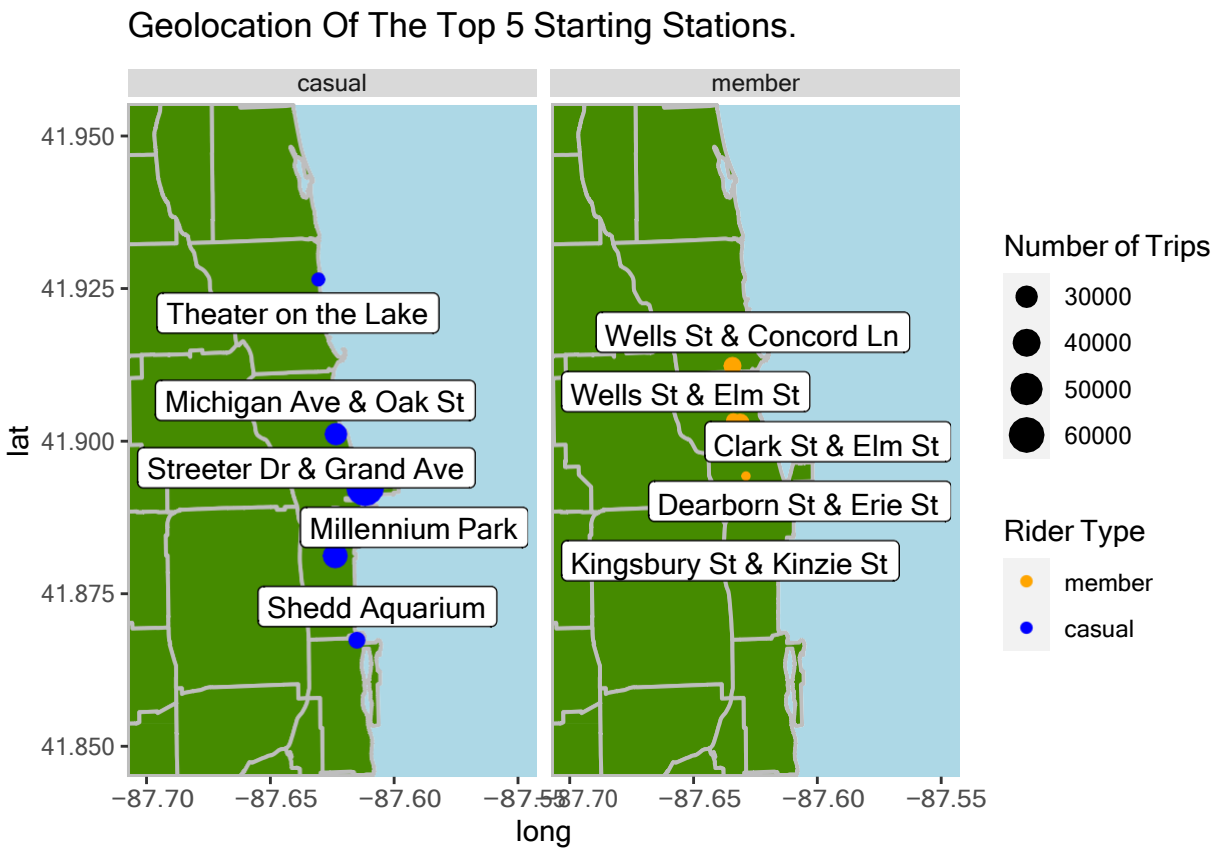
```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\TRICK\Downloads\Boundaries - Community Areas (current)", layer: "geo_export_8d4cbb
## with 77 features
## It has 9 fields
```

```
chi_df = fortify(chicago_map)
```

```
## Regions defined for each Polygons
```

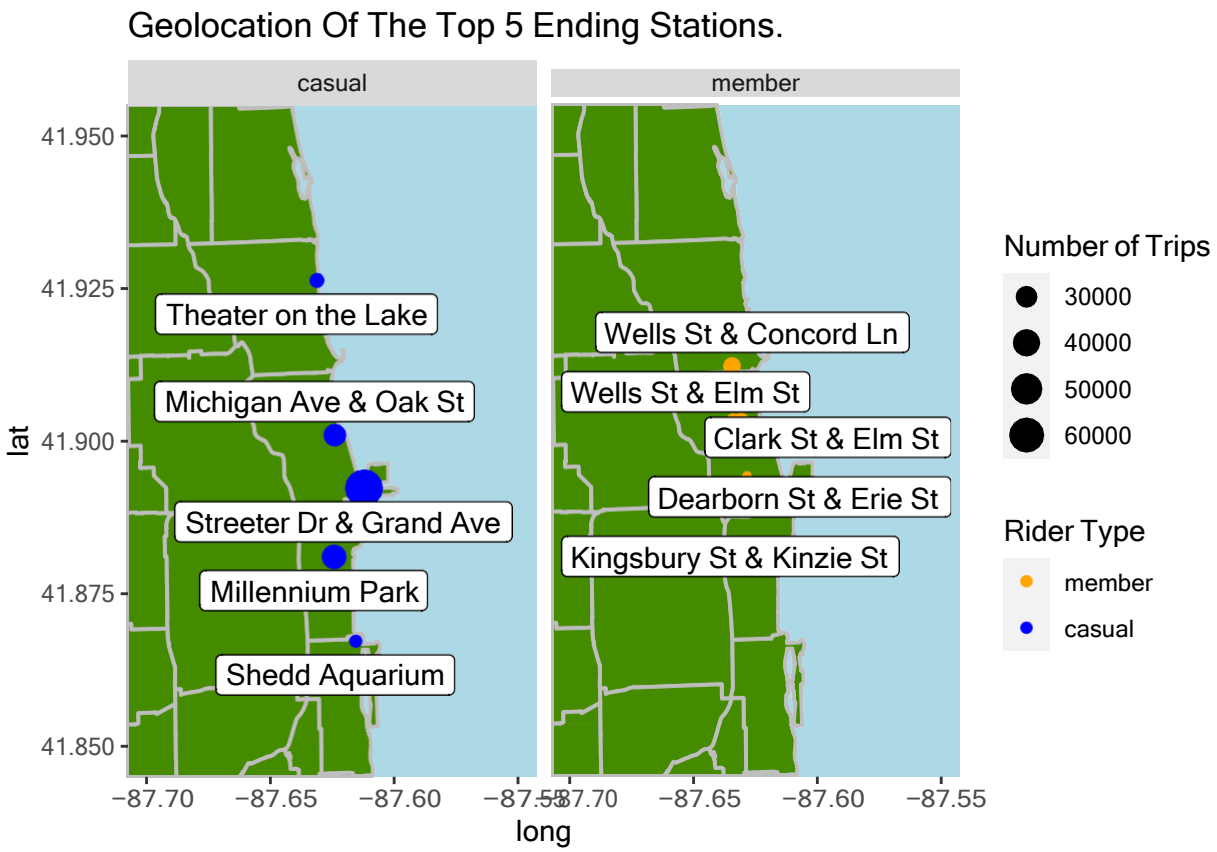
Plotting the start station geolocations.

```
ssgmap <-ggplot() +
    geom_polygon(data = chi_df, aes(x = long, y=lat , group = group), colour = "grey",
    fill = "chartreuse4", size = .7) +
    geom_point(data = start_geo,
             aes(x = Starting_Longitude, y = Starting_Latitude, size = Num_Trips, color = member_casual
             alpha = 1) +
    geom_label_repel(data = start_geo,
                   aes(x = Starting_Longitude, y = Starting_Latitude, label = Start),
                   box.padding   = 0.25,
                   point.padding = 0.65,
                   segment.color = "gray50") +
  scale_colour_manual(values=c(member = "orange",  casual= "blue"))+
  facet_wrap(~member_casual) +
  labs(title = "Geolocation Of The Top 5 Starting Stations.", size = "Number of Trips",
       color = "Rider Type") +
  coord_cartesian(xlim = c(-87.7, -87.55), ylim = c(41.85, 41.95))+
  theme(panel.background = element_rect(fill = "lightblue")) +
        theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
ssgmap
```



Plotting the end station geolocations.

```
esgmap <- ggplot() +
    geom_polygon(data = chi_df, aes(x = long, y=lat , group = group), colour = "grey",
    fill = "chartreuse4", size = .7) +
  geom_point(data = end_geo,
             aes(x = Ending_Longitude, y = Ending_Latitude, size = Num_Trips, color = member_casual),
             alpha = 1) +
  geom_label_repel(data = end_geo,
                   aes(x = Ending_Longitude, y = Ending_Latitude, label = End),
                   box.padding   = 0.25,
                   point.padding = 0.65,
                   segment.color = "gray50") +
  scale_colour_manual(values=c(member = "orange",  casual= "blue")) +
  facet_wrap(~member_casual) +
  labs(title = "Geolocation Of The Top 5 Ending Stations.", size = "Number of Trips",
       color = "Rider Type") +
  coord_cartesian(xlim = c(-87.7, -87.55), ylim = c(41.85, 41.95)) +
    theme(panel.background = element_rect(fill = "lightblue")) +
    theme(panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank())
esgmap
```



SQL Querie to find the mode

```
mode_t <- sqldf("SELECT day_of_week, member_casual, COUNT(day_of_week) AS Total
                 FROM df
                 GROUP BY member_casual, day_of_week
                 ORDER BY day_of_week DESC", method = "auto")
```

Giving the values back to normal

```
mode_t$day_of_week[mode_t$day_of_week == "1"] <- "Sunday"
mode_t$day_of_week[mode_t$day_of_week == "2"] <- "Monday"
mode_t$day_of_week[mode_t$day_of_week == "3"] <- "Tuesday"
mode_t$day_of_week[mode_t$day_of_week == "4"] <- "Wednesday"
mode_t$day_of_week[mode_t$day_of_week == "5"] <- "Thursday"
mode_t$day_of_week[mode_t$day_of_week == "6"] <- "Friday"
mode_t$day_of_week[mode_t$day_of_week == "7"] <- "Saturday"
```

##Plotting the Modes

This function locks x axis so that it does not get sorted

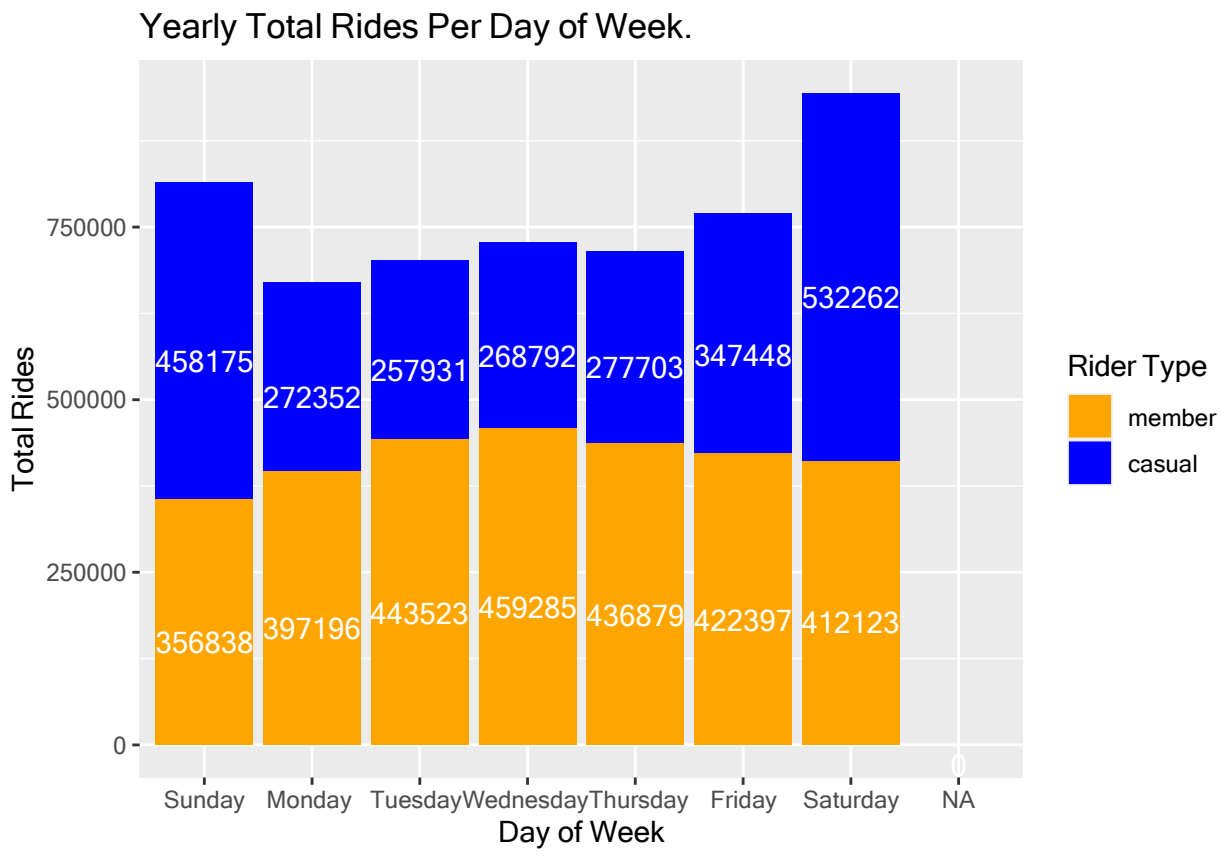
```
mode_t$day_of_week <- factor(mode_t$day_of_week, levels = rev(unique(mode_t$day_of_week)), ordered=TRUE
```

This function finds the sum of casual and member riders, to be used to plot labels

```
mode_t <- mode_t %>%
  arrange(day_of_week, rev(member_casual)) %>%
  group_by(day_of_week) %>%
  mutate(GTotal = cumsum(Total) - 0.5 * Total)
```

A stacked bar plot with the yearly modes for all riders

```
Mode_plot <- ggplot(data = mode_t, aes(x = day_of_week, y = Total, fill = member_casual)) +
             scale_fill_manual(values=c(member = "orange",  casual= "blue")) +
  geom_col() +
  geom_text(aes(y = GTotal, label = Total), vjust = 1.5, colour = "white") +
  labs(title = "Yearly Total Rides Per Day of Week.", x = "Day of Week",
       y = "Total Rides", fill = "Rider Type") +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))
Mode_plot
```



A query to return results related to rideble types used by members

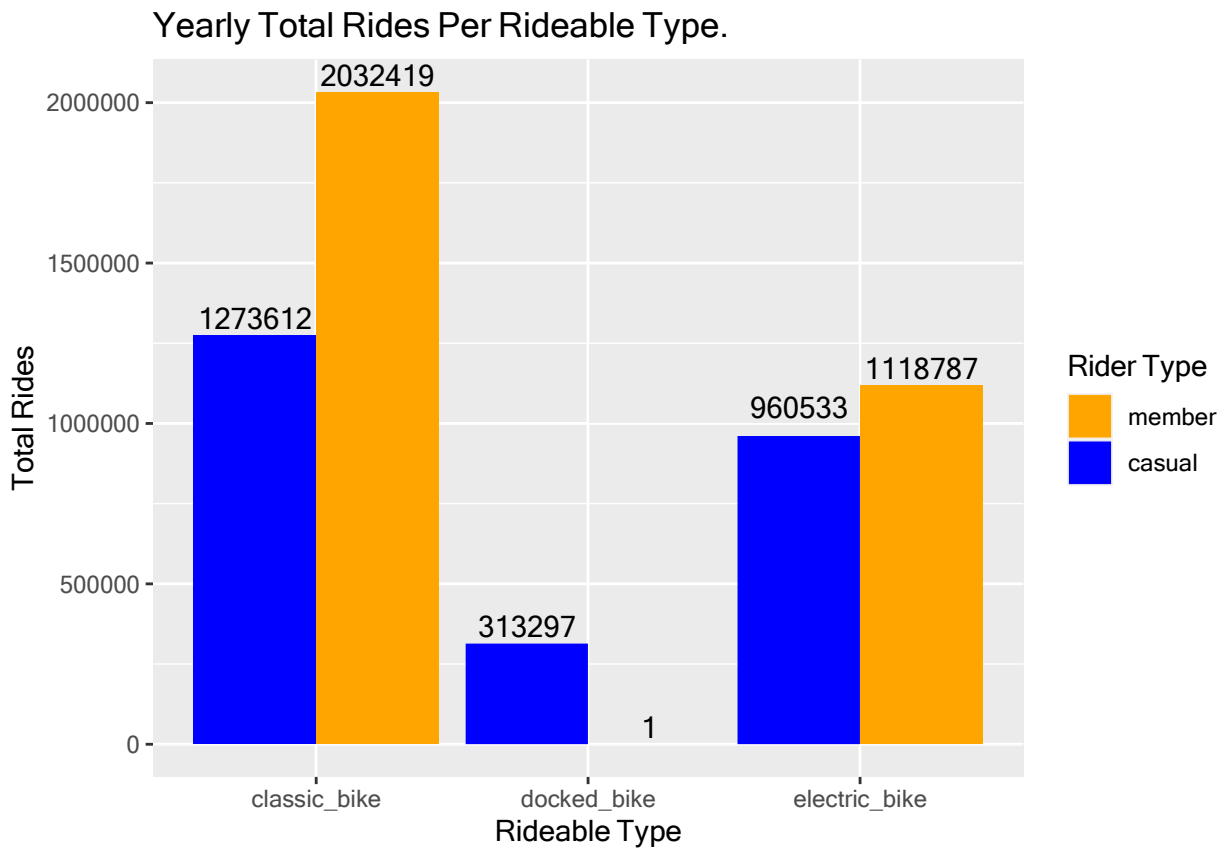
```
bike_df <- sqldf("SELECT rideable_type, member_casual, count(rideable_type) as number_of_uses
                 FROM df
                 GROUP BY member_casual, rideable_type
                 ORDER BY count(rideable_type) DESC", method = "auto" )
```

Changing the names of the rideable type to remove the underscore

```
"Classic Bike"<-bike_df$rideable_type[bike_df$rideable_type == "classic_bike"]
"Docked Bike"<-bike_df$rideable_type[bike_df$rideable_type == "docked_bike"]
"Electric Bike"<-bike_df$rideable_type[bike_df$rideable_type == "electric_bike"]
```

A side by side bar plot

```
bike_plot <- ggplot(data = bike_df, aes(x = rideable_type, y = number_of_uses, fill = member_casual)) +
  scale_fill_manual(values=c(member = "orange",  casual= "blue")) +
  geom_col(position = "dodge") +
  geom_text(aes(label = number_of_uses),  vjust = -0.3 ,colour = "black",
            position = position_dodge(.9)) +
  labs(title = "Yearly Total Rides Per Rideable Type.", x = "Rideable Type",
       y = "Total Rides", fill = "Rider Type") +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))
bike_plot
```



#== # STEP 5: EXPORT SUMMARY FILE FOR FURTHER ANALYSIS #====================================

```
write.csv(df,"C:\\Users\\TRICK\\OneDrive\\Desktop\\GOOGLE\\sort.csv",row.names=FALSE)
```

#END