Untitled

Harshith

3 March 2022

#===================== # 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/CSV DATA OF PAST 12 MONTHS/202101-divvy-tripdata.csv")  
FEB2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202102-divvy-tripdata.csv")  
MAR2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202103-divvy-tripdata.csv")  
APR2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202104-divvy-tripdata.csv")  
MAY2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202105-divvy-tripdata.csv")  
JUNE2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202106-divvy-tripdata.csv")  
JULY2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202107-divvy-tripdata.csv")  
AUG2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202108-divvy-tripdata.csv")  
SEP2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202109-divvy-tripdata.csv")  
OCT2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202110-divvy-tripdata.csv")  
NOV2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202111-divvy-tripdata.csv")  
DEC2021 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202112-divvy-tripdata.csv")  
JAN2022 <- read.csv("C:/Users/TRICK/OneDrive/Desktop/GOOGLE/DA PROJECT 1-CYCLE ANALYTICS/CSV DATA OF PAST 12 MONTHS/202201-divvy-tripdata.csv")

## 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(JULY2021)+nrow(AUG2021)+nrow(SEP2021)+nrow(OCT2021)+nrow(NOV2021)+nrow(DEC2021)+nrow(JAN2022)

#==================================================== # 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,OCT2021,NOV2021,DEC2021,JAN2022))

### 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="geo\_export\_8d4cbb2f-b496-49a4-9ab9-f99b1d9d1c45")

## Warning in OGRSpatialRef(dsn, layer, morphFromESRI = morphFromESRI, dumpSRS  
## = dumpSRS, : Discarded datum WGS84 in Proj4 definition: +proj=longlat  
## +ellps=WGS84 +no\_defs

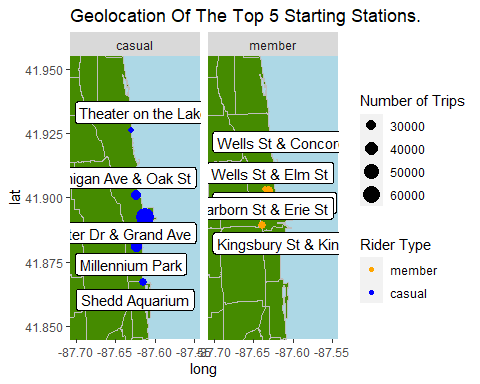
## OGR data source with driver: ESRI Shapefile   
## Source: "C:\Users\TRICK\Downloads\Boundaries - Community Areas (current)", layer: "geo\_export\_8d4cbb2f-b496-49a4-9ab9-f99b1d9d1c45"  
## with 77 features  
## It has 9 fields

chi\_df = fortify(chicago\_map)

## Regions defined for each Polygons

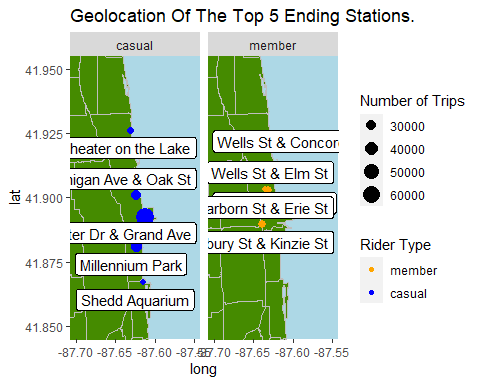
## Plotting the start station geolocations.

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

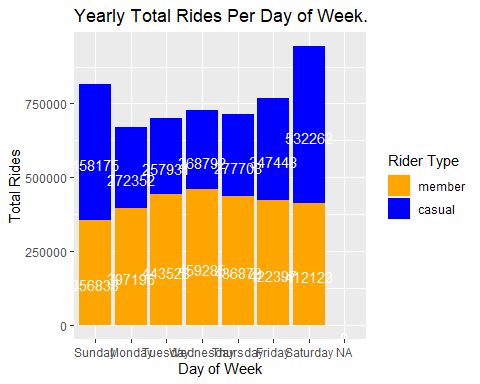
mode\_t$day\_of\_week <- factor(mode\_t$day\_of\_week, levels = rev(unique(mode\_t$day\_of\_week)), ordered=TRUE)

### 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

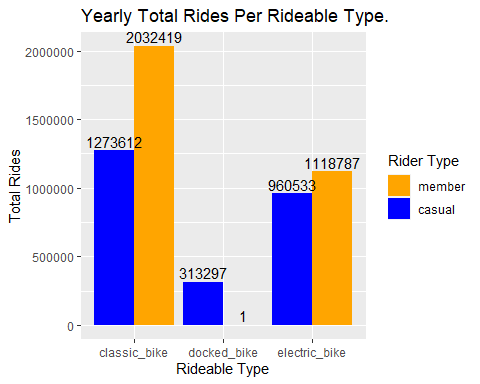
bike\_df <- sqldf("SELECT rideable\_type, member\_casual, count(rideable\_type) as number\_of\_uses  
 FROM df  
 GROUP BY member\_casual, rideable\_type  
 ORDER BY count(rideable\_type) DESC", method = "auto" )

### 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