Case Study: Cyclistic

Question to answer: How do annual members and casual riders use Cyclistic bikes differently?

Problem to solve: How to encourage more casual riders to become members of Cyclistic

Data used: 12 month history of Cyclistic trip data provided by the company, stored on their database by month.

Cleaning process:

1. Highlighted all empty cells using conditional formatting(select all columns, conditional formatting, highlight all cells that contain blanks)
2. Removed all duplicate ride id’s(select entire sheet, or just ride id column, data->remove duplicates)
3. Sorted entire sheet based on member\_casual column
4. Changed format of start and end times to all be in the same date-time format(select cell below header, ctrl+shift+down arrow, right click, format)
5. Aligned all text to the left(select columns->right click->format->alignment->under horizontal, left indent)
6. Autofit column width for all columns

Data processing:

1. Created a column named ride\_length and put in a formula to calculate ride length(=IF(D2=””,””,(D2-C2))) and formatted it into duration format
2. Sorted entire sheet excluding the header row based on casual vs member
3. Created column called day\_of\_week and used =WEEKDAY() command, 1=Sunday, 7=Saturday
4. Created Average ride length column and calculated total average ride time using =AVERAGEIF(N:N,”<>0”), formatted to duration format
5. Found max ride length and most common day for riding for all members
6. Created a pivot table to calculate average ride length for members vs casual riders based on the day, and calculated the number of rides for users by day of the week(pivot table->columns=day\_of\_week, rows=members\_casual, values=ride\_id and ride\_length(click drop down->value field settings->summarize values by average->click number format->time->[h]:mm:ss format)
7. Repeated all Cleaning and Processing steps for all 12 spreadsheets

Data Visualization in R

1. Used command install.packages(“”) for the tidyverse, lubridate, ggplot2, dplyr, and readxl packages, and ran the library function on all of them
2. Used command m1 <- read\_excel("C:/Users/Andrew Tran/Desktop/Processed Trip data/202201-divvy-tripdata.xlsx") for all excel files of cleaned data, naming them m1-12 respectively based on the month the data was acquired and then command pivot1 <- read\_excel("C:/Users/Andrew Tran/Desktop/Processed Trip data/202201-divvy-tripdata.xlsx", sheet = "Jan-2022 pivot table") for all respective pivot tables and their pivot table names
3. Ran str(m1) for m1-m12 to ensure that all the data types are what they are supposed to be, if data type is incorrect used function mutate(m1, column\_name = as.type\_of\_data(column name))
4. Renamed all columns in each data set using function m1 <- rename(m1, trip\_id = ride\_id, bikeid = 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)
5. Removed unnecessary columns using all\_trips <- all\_trips %>%

select(-c(start\_lat, start\_lng, end\_lat, end\_lng))

1. Inspected the new table by using colnames(all\_trips), nrow(all\_trips), dim(all\_trips), head(all\_trips), str(all\_trips), summary(all\_trips)
2. Used table(all\_trips$usertype) to see how many observations were made for casuals and members, also checking to make sure there are only the 2 variables in this column
3. Changed all labels of member and casual to Subscriber and Customer using: all\_trips <- all\_trips %>%

mutate(usertype = recode(usertype, ‘member’ = ‘Subscriber’, ‘casual’ = ‘Customer’))

1. There was a slight error in the Ride\_length column when moving it from excel into R, it for some reason was not just in HH:MM:SS format like it was in excel, all of it had 1899-12-31 attached to it, so I removed the column using all\_trips <- all\_trips %>% select(-c(Ride\_length)) and re-input it in R using all\_trips$Ride\_length <- difftime(all\_trips$ended\_time, all\_trips$start\_time)
2. I also removed the max\_ride\_length, most\_common\_day, avg\_ride\_length columns because they are currently not needed for this analysis and can easily be put back in if needed and i removed the day\_of\_week column because in excel it put in each day as numbers and I am going to reinput it with R to show the actual day of the week.
3. Converted Ride\_length from a Factor to numeric using all\_trips$Ride\_length <- as.numeric(as.character(all\_trips$Ride\_length)) and verified that it was numeric using is.numeric(all\_trips$Ride\_length)
4. Created a new dataset that only has positive non-zero numbers for Ride\_length and the start and end station id’s did not contain HQ QR(removing the times that divvy employees checked out bikes for inspection) using all\_trips\_v2 <- all\_trips[!(all\_trips$from\_station\_name == "HQ QR" | all\_trips$Ride\_length<=0),]
5. Ran summary(all\_trips$Ride\_length) to view the min, mean, median, and max of the dataset
6. Added a date, month, day, year, and day\_of\_week column using

all\_trips\_v2$date <- as.Date(all\_trips\_v2$start\_time),

all\_trips\_v2$month <- format(as.Date(all\_trips\_v2$date), "%m"),

all\_trips\_v2$day <- format(as.Date(all\_trips\_v2$date), "%d"),

all\_trips\_v2$year <- format(as.Date(all\_trips\_v2$date), "%Y"),

all\_trips\_v2$day\_of\_week <- format(as.Date(all\_trips\_v2$date), "%A")

And then re-ordered the days of the week to be in order from sunday->saturday using all\_trips\_v2$day\_of\_week <- ordered(all\_trips\_v2$day\_of\_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))

1. Compared subscribers and customers using aggregate(all\_trips\_v2$Ride\_length ~ all\_trips\_v2$usertype, FUN = mean) and i did this with mean, median, max, and min, then compared their average ride times by day using aggregate(all\_trips\_v2$Ride\_length ~ all\_trips\_v2$usertype + all\_trips\_v2$day\_of\_week, FUN = mean)
2. Created a bar graph using

all\_trips\_v2 %>%

mutate(weekday = wday(start\_time, label = TRUE)) %>% #creates a abbreviated weekday column in the dataframe

group\_by(usertype, weekday) %>% #groups usertypes by weekday

summarise(number\_of\_rides = n(), average\_duration = mean(Ride\_length)) %>% #calculates the number of rides and their average durations

arrange(usertype, weekday) %>% #sorts the group made in the group\_by function

ggplot(aes(x = weekday, y = number\_of\_rides, fill = usertype)) +

geom\_col(position = "dodge") #creates a bar graph labeling the x-axis with weekday, the y-axis with number\_of\_rides, colors the bars by usertype

1. There was a column created with NA value so I re-inspected the dataset and noticed a large amount of NA values, I referenced the original dataset to double check that the NA values were originally there and not the cause of me messing up during the cleaning process, so I made a new dataset omitting all NA values using all\_trips\_v3 <- na.omit(all\_trips\_v2), and recreated the graph with the same function but using all\_trips\_v3
2. Created a new small dataframe with the average ride lengths based on usertype and day of the week using counts <- aggregate(all\_trips\_v3$Ride\_length ~ all\_trips\_v3$usertype + all\_trips\_v3$day\_of\_week, FUN = mean)
3. Exported the new dataframe as a .csv file to be maybe used in other programs for a better data visualization using write.csv(counts, file = 'C:/Users/Andrew Tran/Desktop/Processed Trip data/divvy\_data.csv')