Code

Case Study 1 - Cyclistic Bike Sharing Company

Hide

```
# Package "tidyverse" already installed, hence installing library "tidyverse"

library(tidyverse)
```

Upload csv files (month-wise) of last one year

Hide

```
# Upload csv files (month-wise) of last one year

df202210 <- read_csv("E:/R/trips_data/202210-divvy-tripdata.csv")
```

```
Rows: 558685 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202211 <- read_csv("E:/R/trips_data/202211-divvy-tripdata.csv")
```

```
Rows: 337735 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202212 <- read_csv("E:/R/trips_data/202212-divvy-tripdata.csv")
```

```
Rows: 181806 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202301 <- read_csv("E:/R/trips_data/202301-divvy-tripdata.csv")
```

```
Rows: 190301 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202302 <- read_csv("E:/R/trips_data/202302-divvy-tripdata.csv")
```

```
Rows: 190445 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202303 <- read_csv("E:/R/trips_data/202303-divvy-tripdata.csv")
```

```
Rows: 258678 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202304 <- read_csv("E:/R/trips_data/202304-divvy-tripdata.csv")
```

```
Rows: 426590 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202305 <- read_csv("E:/R/trips_data/202305-divvy-tripdata.csv")
```

```
Rows: 604827 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202306 <- read_csv("E:/R/trips_data/202306-divvy-tripdata.csv")
```

```
Rows: 719618 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202307 <- read_csv("E:/R/trips_data/202307-divvy-tripdata.csv")
```

```
Rows: 767650 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202308 <- read_csv("E:/R/trips_data/202308-divvy-tripdata.csv")
```

```
Rows: 771693 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Hide

```
df202309 <- read_csv("E:/R/trips_data/202309-divvy-tripdata.csv")
```

```
Rows: 666371 Columns: 13── Column specification ────────────────────────────────────────────────────────
────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_name, end_station_id, membe...
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.
```

Compare column names

Hide

```
col_names<-cbind(colnames(df202210),
                 colnames(df202211),
                 colnames(df202212),
                 colnames(df202301),
                 colnames(df202302),
                 colnames(df202303),
                 colnames(df202304),
                 colnames(df202305),
                 colnames(df202306),
                 colnames(df202307),
                 colnames(df202308),
                 colnames(df202309))
```

Display & analyse column names

Hide

```
knitr::kable(col_names)
```

ride_id	ride_id	ride_id	ride_id	ride_id	ride_id	ride_id	ride_id
rideable_type	rideable_type	rideable_type	rideable_type	rideable_type	rideable_type	rideable_type	rideable_type
started_at	started_at	started_at	started_at	started_at	started_at	started_at	started_at
ended_at	ended_at	ended_at	ended_at	ended_at	ended_at	ended_at	ended_at
start_station_name	start_station_name	start_station_name	start_station_name	start_station_name	start_station_name	start_station_name	start_station_nar
start_station_id	start_station_id	start_station_id	start_station_id	start_station_id	start_station_id	start_station_id	start_station_id
end_station_name	end_station_name	end_station_name	end_station_name	end_station_name	end_station_name	end_station_name	end_station_nan
end_station_id	end_station_id	end_station_id	end_station_id	end_station_id	end_station_id	end_station_id	end_station_id
start_lat	start_lat	start_lat	start_lat	start_lat	start_lat	start_lat	start_lat
start_lng	start_lng	start_lng	start_lng	start_lng	start_lng	start_lng	start_lng
end_lat	end_lat	end_lat	end_lat	end_lat	end_lat	end_lat	end_lat
end_lng	end_lng	end_lng	end_lng	end_lng	end_lng	end_lng	end_lng
member_casual	member_casual	member_casual	member_casual	member_casual	member_casual	member_casual	member_casual

All column are same, therefore merging all files into one large data. Name this new data as “all_trips”

Hide

```
# All column are same, therefore merging all files into one large data
# Name this new data as "all_trips"

all_trips <- bind_rows(df202210,
                       df202211,
                       df202212,
                       df202301,
                       df202302,
                       df202303,
                       df202304,
                       df202305,
                       df202306,
                       df202307,
                       df202308,
                       df202309)
```

Inspect summary of new data

Hide

```
# Inspect summary of new data

summary(all_trips)
```

Check “member_casual” column. Is there any third name except these two?

Hide

```
# Our analysis depends upon annual members and casual riders,
# therefore, checking if all the entries are "member" and "casual" only
unique(all_trips$member_casual)
```

Adding columns that list the date, month, day, and year of each ride

Hide

```
# Adding columns that list the date, month, day, and year of each ride
# This will allow us to aggregate ride data for each month, day, or
# year, before completing these operations we can only aggregate
# at the ride level

all_trips$date <- as.Date(all_trips$started_at)
all_trips$month <- format(all_trips$date, "%m")
all_trips$day <- format(all_trips$date, "%d")
all_trips$year <- format(all_trips$date, "%Y")
all_trips$day_of_week <- format(all_trips$date, "%A")
all_trips$hour <- format(all_trips$date, "%H")
```

Calculate ride length for each ride in seconds

Hide

```
# Now calculating ride_length for each ride in a new column

all_trips$ride_length <- difftime(all_trips$ended_at, all_trips$started_at)
```

Convert ride_length into numeric

Hide

```
# Converting ride_length into numeric

all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
```

Check summary of ride_length column

Hide

```
# Check summary of this column
summary(all_trips$ride_length)
```

Hide

```
# There are some negative and some greater than a day values in ride_length, seems # some error
# therefore, eliminating such values, considering minimum ride_duration between
# two stations is 300 sec and maximum is one day
# Giving this data a new name, so that raw data is saved for reference.

all_trips_v2 <- subset(all_trips, ride_length > 299 & ride_length <86400)
```

Descriptive Analysis

Find mean, median, max, min etc of Annual members and Casual riders

Hide

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean)
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = median)
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = max)
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = min)
```

Hide

```
# To See the average ride time by each day for members vs casual users

aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

Name of Days are not in sequence. Fix this

Hide

```
# Noticed that the days of the week are out of order. Let's fix that.

all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursda
y", "Friday", "Saturday"))
```

Hide

```
# Recheck

aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

Hide

```
# analyze ridership data by type and weekday
# creates weekday field using wday()
# groups by usertype and weekday
# calculates the number of rides and average duration
# calculates the average duration

all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n() ,
  average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)
```

Visualization

Hide

```
# Let's visualize the number of rides by rider type
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") + labs(title = "User Type on Weekdays By Rides")
```

Hide

```
# Create a visualization for average duration
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") + labs(title = "User Type on Weekdays By Duration")
```

Hide

```
all_trips_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, month)  %>%
  ggplot(aes(x = month, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Monthly Average Ride Duration")
```

Hide

```
all_trips_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, month)  %>%
  ggplot(aes(x = month, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Monthly Number of Rides By User Type")
```

Find the stations where Casual riders are maximum. Preparing and Processing data further for this.

Hide

```
# Eliminate all rows where start_station_name is null

all_trips_v3 <- all_trips_v2 %>%
  drop_na(start_station_name)
```

Hide

```
# Eliminate all rows where end_station_name is null

all_trips_v3 <- all_trips_v3 %>%
  drop_na(end_station_name)
```

Hide

```
# Group the data by member_casual and start_station_name, and count the number of  # rides
# to identify the stations where casual riders prefer bike ride
all_trips_v3_grouped <- all_trips_v3 %>%
  group_by(member_casual == "casual", start_station_name) %>%
  summarise(num_of_rides = n())
```

Hide

```
# Filter the data to only keep the top 10 stations with the highest number of rides

df_top10 <- all_trips_v3_grouped %>%
  top_n(10, num_of_rides)
```

Hide

```
# Graph the data using a bar plot, with member_casual as the fill color

ggplot(df_top10, aes(x = start_station_name, y = num_of_rides, fill = "casual")) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Start Station Name", y = "Number of Rides", title = "Number of Rides from Start Station")
```

Hide

```
# Group the data by member_casual and start_station_name, and count the number of rides to identify
# the most selected destiny
all_trips_v3_grouped_2 <- all_trips_v3 %>%
  group_by(member_casual == "casual", end_station_name) %>%
  summarise(num_of_rides = n())

# Filter the data to only keep the top 15 stations with the highest number of rides
df_top10 <- all_trips_v3_grouped_2 %>%
  top_n(10, num_of_rides)

# Graph the data using a bar plot, with member_casual as the fill color
ggplot(df_top10, aes(x = end_station_name, y = num_of_rides, fill = "casual")) +
  geom_bar(stat = "identity", position = "dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Start Station Name", y = "Number of Rides", title = "Number of Rides for End Station")
```