Appendix

Bike-Share Case Study

Prepare

Finding a processing and analysis tool

R and RStudio were not my first tools of choice to analyze the bike-share dataset. Excel is more familiar to me (and to most people). And SQL is *the* querying language. Below is the list of steps I took to try and use Excel, and SQL - but in the end R was the best choice for me.

Excel:

1. All 12 files were opened and saved as individual Excel spreadsheets.
2. Opened all 12 files with excel (had to select "Don't Convert") and saved them in a new working excel folder. This took quite a while to open, not convert, and save as 12 excel files.
3. The size of the monthly files varied from 14MB to 73MB. In order to quickly analyze the structure, I focused the initial EDA on the smallest file (Jan 2024).
4. The Jan 2024 file has a total of 13 columns and 144,874 rows.
5. The columns that will not likely be needed for analysis include the [*ride_id*] (each row is a unique ride after all), [*start_station_id*] and [*end_station_id*] (I don't suspect that I will need this info for my analysis). The other columns may prove useful in establishing where, when, what type of bike was rented, and what type of user rented the bike).
6. [*rideable_type*] is a column that distinguishes if a bike is an electric bike or a classic bike. There are no blank rows in this column for this month.
7. [*member_casual*] is a column that categorizes the rider. There are no blank rows in this column for this month.
8. This quick summary was performed on **one** of the twelve monthly datasets from the past year. Even though it was the smallest dataset, it still took minutes for Excel just to open the file, not to mention performing basic sorting and filtering functions.
9. Conclusion: It would be inefficient to perform such a summary on the eleven larger files, and merging the CSV files into one CSV file would crash Excel when attempting to open it. Another tool must be used.

BigQuery:

1. I attempted to load the CSV files into Google's BigQuery to analyze them with SQL, but the 100MB size limit per file prevented half of the files from loading.

2. I next attempted to create a Google Cloud Storage bucket to load the data into (which could hold that much data and then be consumed by BigQuery), however I would have had to pay Google to host my large datasets. I'm hoping to avoid paying money to process and analyze this dataset… if possible.
3. Conclusion: Unless I want to pay to have Google host my large CSV files, another tool must be used.

RStudio (desktop):

1. Using the combined CSV file, I successfully loaded the data into a dataframe using readr's function read_csv().
2. It looks like RStudio can handle the size of the data, let's explore and clean up the data!

Problems with the data

There were a handful of problems I came across while preparing, processing, and analyzing the data:

1. While calculating the most popular start and stop stations, I received duplicate stations because some stations had differing lat/long values so they were counted independently. I solved for this by manually merging the rows in the casual rider dataset (keeping the lat/long for the highest count but summing the count values into 1 row for the start_station_name) but I didn't do the same process for the am dataset (there were 10 duplicate stations).
    a. This is a good time to mention that one of the biggest issues with the data is that the lat/long is not tied to a **station** - but to a ride.
    b. **Recommendation**: In addition to ride data, there should be a second table with station data that is publicly accessible. This would have allowed me to join the station table with the ride table in order to get the "official" lat/long of the station for calculation purposes.
2. There are a number of trips that had start trip info, but no end trip info. Maybe the bikes were never returned to a station? Maybe the bikes were lost, stolen, damaged? Might be worth asking the tech department what would cause end trip information to be null.
3. There were a handful of rides (~1,500) that had end times before their start times. This resulted in a negative trip duration when I calculated a new column. This could have been an error in data processing before being loaded onto the Divvy website. I decided to remove these records for the ride length histograms only, and left them in for all other analysis.
    a. NOTE: The removal of the negative duration rides only changed the average ride length by 1 second, and the median ride length change was 2 seconds.

Share

Making a map with ggmap in RStudio

I tried making a map with ggmap in RStudio. I was able to create a map using the code below, but opted for the more popular - and dynamically capable - Tableau app.

```
# Making a map in RStudio
```

```
## Register Google API key
# TODO - while writing up: Make sure to mention that I had to go onto Google Cloud Console, generate
# an API key, enable the map services, and restrict the key for security
register_google(key = "API_KEY_HERE")

## Get the map using Google as the source
chicago_map <- get_map(location = c(lon = -87.623177, lat = 41.881832), zoom = 11, source =
"google", maptype = "terrain")

## Map the cr data
ggmap(chicago_map) +
  geom_point(data = station_summary_cr,
             aes(x = start_lng, y = start_lat),
             color = cr_color,
             size = 2) +
  labs(title = "Chicagoland Area with Basemap")
```