Objective

Conduct statistical analysis on the stays based on filters chosen by the user to provide him with a user-friendly debrief

Airbnb Web Scraping

Approach:

- Scraping Airbnb using Python through on-demand user queries by tapping on their API's
- Filters used:
  - City
  - Country
  - Check-in
  - Check-out
  - Adults
  - Children
  - Infants
  - Minimum Price
  - Maximum Price
  - Pages to scrape
- A full-fledged dataframe displaying a summarized output for the user's search

Challenges

- Finding the API and nailing the main key carrying all the information (Multiple Nests)
- <Response 400> as soon as I called the Request URL -> Had to update my Request Headers
- Outputting most fields was simple by flattening the json file at the correct key level, however 2 fields (Picture URLs and Nightly Rate) were nested 3 levels more
    - Used a list comprehension to call them and turning them into a DataFrame
- Outputs with pages>(1) weren't concatenating properly
    - After thorough inspection on the request urls and request headers, I realized that I had two keys: Federated Search ID and Source output that were changing continuously after each session
    - I used the URL Decoder website to understand their context within the URL
    - After numerous trials, I realized that I had to drop them to get the required output

Next Step

- Conduct an exhaustive scraping on all stays available at Airbnb and do a supply-demand study to see which locations are matching market's need and develop a go-to-market strategy to mitigate such deficiency