**GOAL: compare and contrast Hotel and Airbnb guest reviews in the US, segmented out by State.**

*#1 The sources of data that you will extract from.*

Hotels

https://www.kaggle.com/datafiniti/hotel-reviews#Datafiniti\_Hotel\_Reviews.csv

This is a list of 1,000 hotels and their reviews provided by Datafiniti's Business Database. The dataset includes hotel location, name, rating, review data, title, username, and more.

Airbnb

https://public.opendatasoft.com/explore/dataset/airbnb-listings/information/?disjunctive.host\_verifications&disjunctive.amenities&disjunctive.features

Inside Airbnb is an independent, non-commercial set of tools and data that allows you to explore how Airbnb is really being used in cities around the world.

*#2 The type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).*

- Reading database using Python Pandas.

- Ensuring delimiters and character encoding are aligned.

- Testing available data columns

- Selecting target data columns for subsequent merging: filtering, renaming, dropping.

- Normalizing reviews to match scale 1-5

- Unlike the AirBNB data, the data source we had for the hotels was a list of individual reviews. So the goal was not only to clean up the data, but also to calculate the average rating of each hotel so that it matches the AirBnB data.

- To achieve this, we made two dataframes grouping the data by name. One grouped by mean to get the average rating and the other by max to get a single field for the text values. Lastly, we made a dataframe that combined the needed fields.

- The results were still very messy, with hotels that had no ratings, rows with NaN values, and fields with incorrect data. We were able to use dropna and loc to remove the missing values but for the incorrect ones we needed additional steps. We created helper columns returning the length of row using lambda functions to help identify bad data (ie. Zip codes that weren’t 5 digits). We could then pull loc functions to delete those rows. Lastly, we loaded the data into MongoDB

*#3 Load: the final database, tables/collections, and why this was chosen.*

- Selected overlapping data points

Name Address, City, State, Zip, Type, Average Rating

These will let us have some hotel/Airbnb identifiers and corresponding rating.

- Loading cleaned python data into MongoDB

-Creating database connection.

-Creating database vacation

-Creating collection Accommodations

*#4 Possible analysis drawn from curated database.*

States where Airbnb review is higher Hotel

States where Hotel review is higher than Airbnb

Top 5 states in terms of review count

Top 5 states in terms of ratings

Zip code

*#5 Creating an API app for users to easily query curated data by entering two query fields.*

* Used Flask to call on MongoDB database.
* Issues: JSON output was giving us some issues due to syntax discrepancy; we were able to adjust dictionaries for correct API output.