**Airbnb Popularity Analysis**

*Documentation*

By **0% Risk Team**

**Zhexin Hu** (zhexinh), **Yijing Lin** (yijingli), **Lingqin Yang** (ylingqin), **Chengyu Wu** (chengyu2)

**Content**

[searchPageFetch.py 1](#_Toc58583267)

[listingcleaning.py 2](#_Toc58583268)

[nearby\_search.py 3](#_Toc58583269)

[nearby nearby\_analysis.py 3](#_Toc58583270)

[nearby hotmap\_plot.py 3](#_Toc58583271)

[bys\_rating\_describe.py 3](#_Toc58583272)

[analysis on price, superhost, amenities, and response rate.py 3](#_Toc58583273)

[crime\_and\_listing.py 4](#_Toc58583274)

# searchPageFetch.py

* **Description:** This module is used to fetch content from search pages of Airbnb.com for each city. Using Selenium, BeautifulSoup, and json, this module can return information of each listing shown on a search page. It would also find the URL of the next page and self-navigate to turn pages. On the highest level, a progress management CSV file can be used by this module to track and resume collecting progress in case of functionality or connectivity failures.
* **Packages:** time, os, selenium, bs4, json, pandas, numpy, random.
* **get\_page\_source(url):**
  + input parameter: url (string).
  + Return: soup (soup object)
  + Open up a headless, no-window, and no-image browser to load and return page source code.
* **Get\_json(soup):**
  + Input Parameter: soup (soup object)
  + Normalize json content to dictionary
* **Get\_info(js):**
  + Input Parameter: js (dictionary)
  + Find useful information from json dictionary, including id, name, latitude, longitude etc.
* **Get\_nextpageurl(soup):**
  + Input Parameter: soup (soup object)
  + Find link to the next page, return None if not applicable.
* **Get\_progress():**
  + Built-in input: progress (dataframe from CSV)
  + Locate the current progress, return the URL of the last stop, generate new URL if not applicable, return 1 if a neighborhood is finished, return 0 if all neighborhoods are finished.
* **Main():**
  + No input
  + Call get\_progress() to locate current progress, call get\_page\_source() using the saved or generated url. Then call get\_json(), get\_info() and get\_nextpageurl() to decode and extract specific data. Loops are controlled by the dataframe from the progress managing CSV file.

# listingcleaning.py

* **Description:** Calculate Bayesian average rating from original rating data, merge response rate data from CSV downloaded from insideairbnb.com, and correct data type of the master dataset.
* **Packages:** numpy, pandas.
* **bayesian\_average():**
  + Built-in input from the pre-cleaned master dataset (e.g. listings\_clean\_copy.csv)
  + Built-in output to save change to file.
  + Calculate Bayesian average rating based on the original rating using the formula below:

Where M equals to the median of all the review count and A equals to the mean of all the rating.

* **merge\_host\_response\_data():**
  + Built-in input from the pre-cleaned master dataset (e.g. listings\_clean\_copy.csv)
  + Built-in output to save change to file.
  + Merge host response data (CSV) using listing id as “primary key”. This data is obtained by insideairbnb.com from the detail page (not search page) of each listing, where we did not have enough time and resource to perform scraping.
* **data\_type\_correction():**
  + Built-in input from the pre-cleaned master dataset (e.g. listings\_clean\_copy.csv)
  + Built-in output to save change to file.
  + Correct certain data types in some columns.

# nearby\_search.py

* **Packages:** requests, pandas, numpy, ﻿urllib.parse(urlencode).
* **function:** search\_nearby
  + Input: location string("latitude,longitude"), search radius, category of places, how many results.
  + Description: Construct the request url using inputs. Get the JSON response. If the response's status is "ZERO\_RESULTS", meaning no subway/toursit attractions nearby, set the corresponding item in list to 0, otherwise 1.
  + Output: A list recording whether there is subway/tourist attractions nearby for each listing.
* **main**
  + Read from the listings csv file.
  + Loop through each location by (latitude,longitude) and use search\_nearby to find nearby subway/tourist attractions nearby.
  + Add the nearby subway/tourist attractions information back to csv file.

# nearby nearby\_analysis.py

* **Packages:** pandas, seaborn, ﻿matplotlib.pyplot(plt).
* **main**
  + Read from the listings csv file of rating, subway and tourist attractions data.
  + Plot box plot of rating~subway and rating~tourist attraction.

# nearby hotmap\_plot.py

* **Packages:** ﻿geopandas, ﻿shapely.geometry (point, polygon), ﻿mapclassify.
* **main**
  + Read from the listings csv file and clean data that’s with NAN rating.
  + Read from NYC borough shape file.
  + Set coordinate system.
  + Combine latitude and longitude into points.
  + ﻿Create figure and axes, assign to subplot
  + Geoplot with categorical/quantiles.

# bys\_rating\_describe.py

* **Packages:** matplotlib.pyplot, pandas
* **hist\_graph:**
  + Read from listings csv file
  + Create the histogram plot of avgRating and bysAvgRating

# analysis on price, superhost, amenities, and response rate.py

* **Packages:** matplotlib.pyplot, pandas, seaborn
* **desc\_price**
  + Create the histogram plot and scatter plot of listing price against Bayesian average rating
* **superhost\_amenity\_response** 
  + Calculate the total number of amenities
  + Convert host response information to float
  + Covert isSuperhost from integer (1, 0) to string (Yes, No)
  + Create boxplot of number of amenities to Bayesian average rating
  + Create boxplot of number of amenities to Bayesian average rating under superhost or not
  + Create boxplot of superhost
  + Drop nan value in response rate
  + Create scatter plot of response rate to Bayesian average rating
  + Create correlation heatmap of attributes

# crime\_and\_listing.py

* **Description:** This file is used to clean the crime data, modify the cleaned the listing data, merge both data, compute the crime rates for each borough in NYC, and perform exploratory data analysis of crime rate, room types, number of bedrooms/bathrooms and Bayesian average rating.
* **Packages:** pandas, numpy, matplotlib, seaborn
* **Crime\_clean\_and\_merge():**
* reads in three csv files, including crime data, cleaned listing data, and inside airbnb data, which is used to update the cleaned borough information for each listing
* perform data cleaning and merging (converting data types, removing NA, etc)
* creates new column as population and compute the crime rates for each borough
* prints a summary of the final dataset
* Output: a final dataset that can be used to perform exploratory data analysis
* **Analysis\_on\_crime\_roomtype\_roomno():**
* Visualize the distribution of number of listings, price, and Bayesian average rating
* Explore the relationships among crime rates, number of bedrooms/bathrooms and Bayesian average rating
* Output includes the following plots:
* boxplot of crime rates to Bayesian average rating by borough
* bar plot of the distribution of number of listings by borough
* bar plot of the distribution of the Bayesian average rating by borough
* bar plot of price by borough, boxplot of roomtype to Bayesian average rating
* boxplot of roomtype to Bayesian average rating by borough
* boxplot of number of bedroom to Bayesian average rating
* scatter plot of number of bedroom to Bayesian average rating
* boxplot of number of bathroom to Bayesian average rating
* scatter plot of number of bathroom to Bayesian average rating
* **Main()**
* call Crime\_clean\_and\_merge() and Analysis\_on\_crime\_roomtype\_roomno()