While examining New York City’s AirBNB vacancy rates, I thought it would be a good idea to see if there were any neighborhoods with a lot of listings and high availability rates, with the idea that these listings could be used in an experiment to try to generate incremental revenue for the company. Specifically, AirBNB could monitor session data to send targeted emails or advertisements to users who recently viewed NYC AirBNB listings (last 30 days) but did not complete a booking. These users would receive an email or advertisement promoting a house/apartment in one of the defined target neighborhoods with many listings and high vacancy rates with an offer to reduce the fee AirBNB collects by 50%. If AirBNB normally takes a 10% cut, the promotion would offer to reduce that to 5%.

The promotion will be valid for up to 30 days (we should do a check on mean/median conversion time from first looking at the listings to finished booking and make sure 30 days is a viable time frame).

The first step to carry out this experiment is to use the data to identify candidate neighborhoods. See code below.

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

import seaborn as sns

%matplotlib inline

df = pd.read\_csv('C:\\Users\\ryan\\Desktop\\Thinkful DS Sample Data - Main Course\\Unit 1\\AirBNB Data\\listings\_summ.csv')

#df.head()

#Get average availability and count by neighborhood

df2 = df.groupby('neighbourhood', as\_index = False).mean().sort\_values('availability\_365', ascending = True)

df3 = df.groupby('neighbourhood', as\_index = False).count().sort\_values('availability\_365', ascending = True)

#Combine these two dataframes into one

df4 = df2

df4['Listings\_Count'] = df3['id']

#Target less desirable neighborhoods to use for our campaign. Let's use neighborhoods with >50 listings and #an average yearly availablility >200 days.

target\_hoods = df4[(df4.Listings\_Count > 50) & (df4.availability\_365 > 200)].sort\_values('Listings\_Count', ascending = False)

#Rank-order the target neighborhoods based on revenue opportunity.

target\_hoods['rank\_index'] = target\_hoods['price'] \* target\_hoods['availability\_365']

target\_hoods.sort\_values('rank\_index', ascending = False)

#Now we have an ordered list of neighborhoods to discount for our experiment.

#Visualize the target neighborhoods

#Get subset of neighborhoods from main listing file that match the neighborhoods in our target list.

df\_hood\_sub = df.merge(target\_hoods, how='left', on='neighbourhood')

#Add segment for hue in visualization

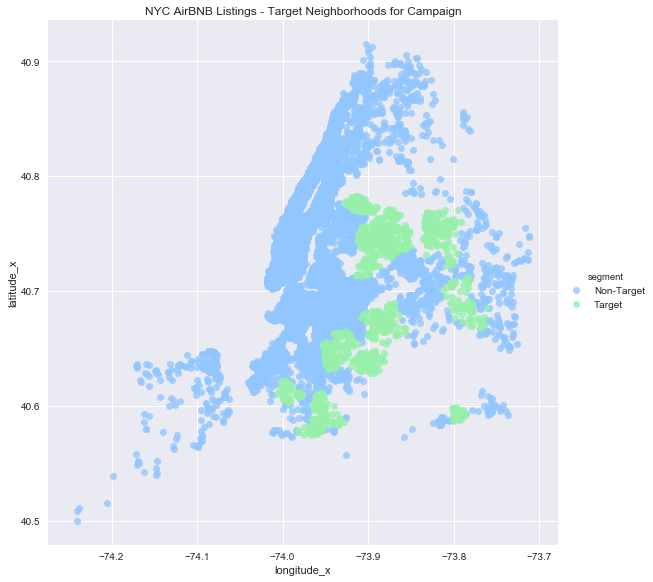
target\_hoods\_list = target\_hoods.neighbourhood.unique()

df\_hood\_sub['segment'] = df\_hood\_sub['neighbourhood'].apply(lambda x: 'Target' if x in target\_hoods\_list else 'Non-Target')

#Plot the results

sns.lmplot("longitude\_x", "latitude\_x", data = df\_hood\_sub, fit\_reg = False, palette = 'pastel', size = 8, hue = 'segment')

plt.title('NYC AirBNB Listings - Target Neighborhoods for Campaign')



The figure above provides a nice spatial visualization of the AirBNB listings by using the latitude and longitude data provided in the dataset.

Roll-out:

With the target neighborhoods now defined, the roll out would proceed as follows:

1. Define the population to be targeted. For this experiment, we will have two separate groups.
   1. Registered users – these users will be targeted via email.
   2. Non-Registered users – these users will be targeted via advertisements on Facebook, Google, etc., using ‘pixel’ information from their browser history, obtained while on AirBNB.

As stated above, eligible users must have searched AirBNB for NYC listings within the last 30 days. Each group will be divided into a test and control sample, 50% in each. Effort should be made to ensure the test and control groups are randomly selected and have similar distributions in terms of the users’ geographical location, prior AirBNB purchase history, site activity levels (such as unique visits in the past 30 days), and NYC listings viewed prior to the experiment (eg. a user who was searching for expensive apartments in Chelsea is unlikely to be tempted by a small discount on an apartment in Queens).

1. Check to make sure these target users are not involved in any other tests or marketing campaigns. Test the infrastructure – make sure that we can actually identify and reach the target users, particularly the non-registered site users that we plan to reach via advertisement. Need to prove that we can effectively reach our audience and in a timely manner.
2. Begin gradual roll-out of emails (Group 1A) and advertising campaign (Group 1B) at the same time to the test populations. This will remove any seasonality affect from the outcome of the experiment amongst the two test groups. The gradual roll-out will begin with 10% of the intended 50% test population. Roll-out will begin with promoted listings starting from the top-ranked neighborhoods (price \* listings, see code above).
3. Track conversion rates on a daily basis for users in the test and control groups over the next 30 days. If there are any issues, we will be able to capture them quickly with daily monitoring.
4. If there are no production, technical, or severe business issues after 7 days, roll out to the remaining test group members. Continue daily monitoring of results for 30 days after promotion offer extended.

Defining Success:

The experiment will be considered successful if the test population generates incremental revenue for the company. Incremental revenue is defined as the 30-day cumulative revenue of the test group minus the 30-day cumulative revenue of the control group.

Secondary metrics to be tracked:

1. How did conversion rates vary between Groups 1A and 1B? Were existing users more likely to ‘bite’ on the discount than unregistered users?
2. Some users likely explored NYC apartments but ultimately decided to visit a different city, and wound up booking a listing there. To what extent was the rate of non-NYC bookings affected over the next 30 days for the test groups?

In evaluating the experiment results, we can use a T-Test to determine if the average incremental revenue per user of the test group is different from the control group. (User revenue data should look fairly normal if there are enough users in the test and control groups, but if the distributions are not normal, use a non-parametric statistical test instead of a T-Test.)