Brief Project Overview

For my project, I will primarily be working with the Food Establishment Inspections dataset from Boston.gov. This dataset contains detailed information regarding Boston food establishment inspections from 12/27/2006 up to 4/19/2015. (It's updated daily.)

I will combine information in this dataset with Yelp user reviews to explore possible relationships between restaurant health code violations and general public opinion of those restaurants. I will obtain these reviews through Yelp's API.

Goals

1. To investigate the possible relationship between number of violations (and the severity of such violations) and Yelp user ratings. I suspect that restaurants with high user ratings will have few to no health code violations, but it is possible that no relationship exists.

2. To investigate the possible relationship between number of violations (or type of violations) and restaurant location. (This goal will involve clustering the data by location and number of violations.)

3. (Just for fun) To investigate hospital food establishment inspections. For example, Massachusetts General Hospital had serious food contamination issues (they got caught having food touch the floor). I'd like to compare hospitals to one another in this respect, and again compare this data to Yelp's data.

Brief Dataset Overview

I downloaded the dataset as a .CSV file, so loaded the dataset into a Pandas dataframe:

```
Int64Index: 316889 entries, 0 to 316888
Data columns (total 26 columns):
BusinessName    316889 non-null object
DBAName         3970 non-null object
LegalOwner      215418 non-null object
NameLast        316889 non-null object
NameFirst       152872 non-null object
LICENSENO       316889 non-null int64
ISSDTTM         316731 non-null datetime64[ns]
EXPDTTM         316587 non-null datetime64[ns]
LICSTATUS       316889 non-null object
LICENSECAT      316889 non-null object
DESCRIPT        316889 non-null object
RESULT          316889 non-null object
RESULTDTTM      311536 non-null datetime64[ns]
Violation       299413 non-null object
ViolLevel       299413 non-null object
ViolDesc        299413 non-null object
VIOLDTTM        299411 non-null datetime64[ns]
ViolStatus      295317 non-null object
StatusDate      131784 non-null datetime64[ns]
Comments        281959 non-null object
Address         316767 non-null object
City            316746 non-null object
State           316258 non-null object
Zip             316054 non-null object
Property_ID     223346 non-null float64
Location        217276 non-null object
```

The dataset unfortunately does not come with any description, but from Googling around, I've determined what majority of these column titles refer to.

	Data Type	Description	Missing Values?
BusinessName	String	Name of the food establishment	No
DBAName	String	Trade name of the food establishment. (DBA = "Doing Business As"). For example, the restaurant "Extreme Pita" on BU campus has its business name listed as "Extreme Pita", but its DBA name listed as "Trustees of Boston University"	Yes
LegalOwner	String	Legal owner of the business. (The legal owner is typically the parent company's name from what I can tell - e.g., ARAMARK)	Yes
NameLast	String	Owner's first name (Could be a person's name OR a company name)	No
NameFirst	String	Owner's last name (if the owner is a company, then this field is blank)	Yes
LICENSENO	int64	The food establishment's business license number	No
ISSDTTM	datetime	Issue date for food license	Yes
EXPDTTM	datetime	Expiration date for food license	Yes
LICSTATUS	String	License status. Takes on 1 of 3 values: {Active, Deleted, Inactive}	No
LICENSECAT	String	Food license category. Takes on 1 of 3 values: {FS,FT,MFW}; MFW = "Mobile Food Walk On" - AKA food truck FS = "Eating & Drinking" FT = "Eating & Drinking W/ Take Out"	No
DESCRIPT	String	Description of license category. Takes on 1 of 3 values: "Mobile Food Walk On", "Eating & Drinking", and "Eating & Drinking W/ Take Out"	No
RESULT	String	Result of inspection. Takes on 1 of 13 possible values: HE_fail = failed inspection Fail = failed inspection Failed = failed inspection HE_pass = passed inspection Pass = passed inspection HE_filed = "Pass w/minor violations" HE_closure = closed HE_FAILNOR = ? HE_hearing = need to attend a hearing HE_NotReq = "not required"? HE_Misc = miscellaneous HE_OutBus = ? HE_TSOP = temporary suspension of permit	No
RESULTDTTM	datetime	Date and time the results were entered	Yes
Violation	String	Type of violation. The violation is given as a code number from here:	Yes

		http://www.cityofboston.gov/isd/health/violationcodes.asp	
ViolLevel	String	Violation level. Can take on one of 4 values: {NaN,*, **, ***}. NaN is reserved for food establishments which have closed.	**Yes**
ViolDesc	String	Violation description	**Yes**
VIOLDTTM	datetime	Date and time of violation	**Yes**
ViolStatus	String	Violation status. Takes on 1 of 3 values: {NaN, Pass, Fail}	**Yes**
StatusDate	datetime	Sometimes information in the database is updated for certain restaurants. StatusDate marks when the data entry was updated.	**Yes**
Comments	String	The health inspector's comments. (e.g., "elevate food 6 inches off floor")	**Yes**
Address	String	Physical address of the food establishment	**Yes**
City	String	These values are just the neighborhoods of Boston (e.g., Roxbury, Roslindale, Mission Hill, etc.)	**Yes**
State	String	State the food establishment belongs to (it's always Massachusetts)	**Yes**
Zip	int64	Zipcode of the food establishment	**Yes**
Property_ID	float64	Unique ID for the property. (Several properties have a Property_ID of 0, so this Property_ID column might not be very meaningful.)	**Yes**
Location	float	Geolocation of the food establishment (stored as a longitude/latitude tuple)	**Yes**

As you can see, this dataset has quite a few missing values across many of its columns. However, we actually do expect missing values for certain columns, for example:

a. **DBAName:** Not every company has a DBA name. DBA names are not required by law.
b. **Legal Owne**r: Some companies (but not all) are legally owned by another company (parent company). That's basically what "legal owner" is used for in this instance.
c. **NameFirst**: This field goes along with the "NameLast" field. For any food establishment, if a person owns the establishment, their first name will show up in this field. If a company owns the establishment, then no name will show up in this field.
d. **Violation/ViolLevel/ViolDesc/VIOLDTTM/ViolStatus**: Not every restaurant has a violation.
e. **StatusDate:** If a health inspector doesn't update the status, then there is no status date.
f. **Comments**: Health inspector's comments. We don't expect every health inspector to leave comments!

For columns that we do *not* expect to have missing data, here is how I plan to work with them:

1. Issue Date (ISSDTTM) and Expiration Date (EXPDTTM)

I don't plan to use either of these columns of data. As I mentioned briefly in my "Goals" section, I am primarily concerned with each restaurant's: (1) geolocation, (2) violation information, and (3) Yelp user reviews. Issue date and expiration date are both irrelevant to my project goals.

2. Address, City, State, Zip, Location

Here, I only plan to work with City and Location. I am interested in "City" because I would like to see if I can rank neighborhoods based on the number of failed health inspections. (Remember that "City" means neighborhood.) I am interested in "Location" so that I can cluster data using geolocations.) Address, State, and Zip I do not plan to use. (In fact, "State" is always Massachusetts, so it's not meaningful at all.)

I plan to use Geopy to fill in missing geolocation entries. Geopy takes an address as an input, then converts that address to a geolocation pair - e.g., (42.3496, -71.0097)

If both the food establishment's geolocation *and* address are missing, I plan to use Yelp's API along with Geopy to help recover the geolocation of the restaurant. (Yelp will hopefully be able to obtain the address of a restaurant, but there certainly are no guarantees.) Fortunately, we're only missing 122 locations, which is a small amount relative to the total number of entries (316,889).

**Currently, there are 217276 of 316889 "location" entries filled in the dataset, which means about 30% of the location entries are empty. However, I do not expect to fill out all of the entries. For example, entry #23322 in the dataset shows up as:

BusinessName	Ben & Jerry's		
DBAName	NaN	Violation	NaN
LegalOwner	NaN	ViolLevel	NaN
NameLast	Boston Scoop Shop Inc.	ViolDesc	NaN
NameFirst	Jason Sweeney	VIOLDTTM	NaN
LICENSENO	68185	ViolStatus	NaN
ISSDTTM	2012-05-31 09:53:58	StatusDate	NaN
EXPDTTM	2011-12-31 23:59:00	Comments	NaN
LICSTATUS	Inactive	Address	NaN
LICENSECAT	FT	City	NaN
DESCRIPT	Eating & Drinking w/ Take Out	State	NaN
RESULT	HE_Pass	Zip	NaN
RESULTDTTM	2010-12-13 08:46:09	Property_ID	NaN

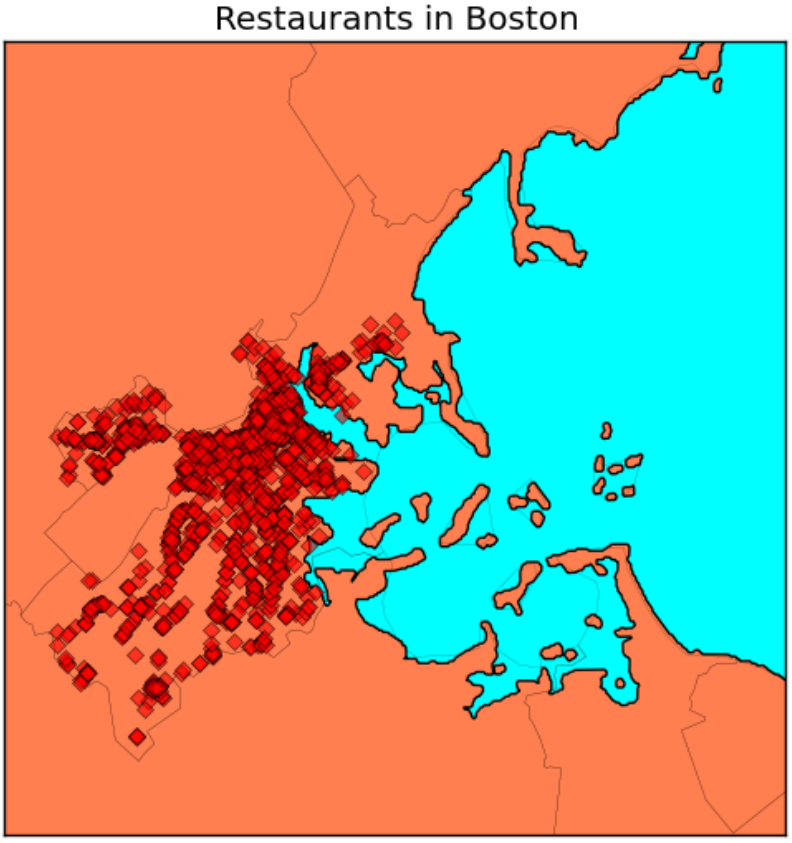
Here, we see there is no address, city, state, zip, etc. information. If we look at ISSDTTM and EXPDTTM, we can tell this business is no longer in operation. These types of entries I will discard.

3. RESULTDTTM

The information here could be useful when compared to violation information, although I don't think it's possible to obtain the missing information here since the only people who input this information are the health inspectors themselves.

First Impressions

It turns out that filling in missing geolocation data takes quite a lot of time (about 2-5 seconds per missing entry), so I haven't yet filled in all of the geolocations at this time. However, here is a map of restaurant locations from the given locations in the original dataset:



We can see that the density of restaurants is highest near the northern part of Boston - which makes sense because the population density in the southern areas of Boston is lower than in the northern areas.

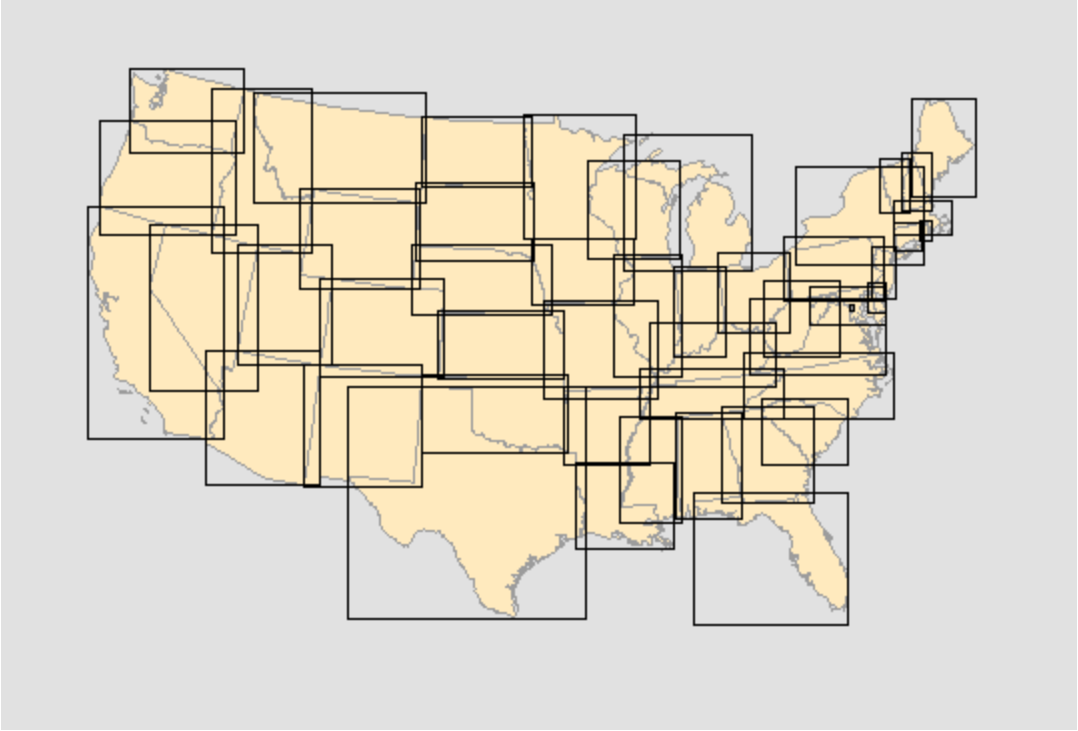
There is no specific area of Boston where location data is missing from. The number of missing entries is given by neighborhood:

BOSTON: 859 EAST BOSTON: 201 DORCHESTER: 139 ALLSTON: 71 SOUTH BOSTON: 62 ROXBURY: 46 BRIGHTON: 55 HYDE PARK: 23 JAMAICA PLAIN: 23 WEST ROXBURY: 23	MATTAPAN: 22 ROSLINDALE: 22 CHARLESTOWN: 18 FENWAY: 7 FINANCIAL DISTRICT: 7 WEST END: 2 Mission Hill: 2 CHINATOWN: 1 CHESTNUT HILL: 1

Immediately, we can tell that some restaurant locations are *not* specified by neighborhood. For instance, many restaurant locations are simply given as "Boston". We also see that one of the

restaurant locations is given as “Chestnut Hill”, which is not a neighborhood in Boston. I will therefore use information gathered from Yelp to help correct the restaurant neighborhoods.

Worst case, if using Yelp to correct the neighborhoods does not work (or only partially corrects the neighborhood data), I will consider the alternative approach of creating minimum-bounding rectangles (MBRs) for each neighborhood. For example, MBRs for each state in the U.S.:



There is an R-Tree package available for Python which I could do this with: https://pypi.python.org/pypi/Rtree/

I can also write an algorithm to compute the MBRs myself. (There are only 23 neighborhoods in Boston.)