Safe Tourism in Boston:
Landmarks, Transportation and Crime

Project Goal:
Our goal is to determine best travel experiences for incoming tourists within Greater Boston Area. Having such an immense area, people may not have their best experiences in their limited time of travel and we wanted to suggest specific areas based on various datasets for the best possible experience.

Data sources Used:
Analyze Boston (*data.boston.gov*)
Boston Maps Open Data (*bostonopendata-boston.opendata.arcgis.com*)
Massachusetts Department of Transportation (*geo-massdot.opendata.arcgis.com*)

Datasets Used:
Boston Neighborhoods (*get_neighborhoods.py*) https://data.boston.gov/dataset/boston-neighborhoods
Crime rate (*get_crimeData.py*)
https://data.boston.gov/dataset/crime-incident-reports-august-2015-to-date-source-new-system
Boston Landmarks Commission (BLC) Historic Districts (*get_landmarks.py*)
http://bostonopendata-boston.opendata.arcgis.com/datasets/547a3ccb7ab443ceaaba62eef6694e74_4
MBTA Bus Stops (*get_busStops.py*)
https://geo-massdot.opendata.arcgis.com/datasets/2c00111621954fa08ff44283364bba70_0
MBTA Station stops (*get_trainStations.py*)
https://geo-massdot.opendata.arcgis.com/datasets/train-stations?geometry=-73.51%2C41.878%2C-69.555%2C42.59

Project Description:
We currently put together the few datasets listed above and transformed the acquired datasets to see which neighborhoods within the Greater Boston Area
- has greater number of landmarks to see
- has better system of public transportation
- has low crime rates.

Algorithm and Analysis:
We needed a way to rate the neighborhoods somehow based on the coordinate data we have collected. The problem is that we have each neighborhood's landmark coordinates, public transportation coordinates and crime coordinates (where crime occurred) but we do not have a way to compare a neighborhood to another neighborhood.

We decided to use K-means method to find cluster of coordinates with positive values (landmark coordinates and public transportation coordinates). The K-means would give us a K number of coordinates where the data is clustered at. This would not let us compare the neighborhoods but based on where each coordinate lies, we may be able to take this information to rate the neighborhoods. We could possibly decide to give the coordinates to the center of the clusters found by the K-mean algorithm for the users to possibly create a better experience.

Secondly, we found the average distance of all features of a neighborhood from its averaged coordinate of its features (if the wording here is confusing, I have described what each file does below; please scroll down to where file name includes *stat*). I would call this algorithm to be somewhat of an scuffed insight to K-means. Based on the resulting averaged distances to averaged coordinates of neighborhood's features, we are able to rank the neighborhoods in a manner (I will refer to the averaged distance value as stats value). The stats value would tell us which town would be better to travel to based on how clustered the features are in each town. Basically lower value means landmarks and transportations are packed tightly together within the neighborhood. This is not comparable directly to the K-means but the stats algorithm gives us a different insight within each town's data of coordinates.

Also, we created different variations for the K-means and stats algorithm for user experience in that transportation coordinates are included or excluded: considering user may use public transportation or just simply ride UBER(or drive their own car)
crime coordinates are included or excluded: considering safety may not be a issue to the user.

stat_landmark.py:
- finds the averaging center point of landmarks based on each neighborhood's landmark coordinates
- then finds the average distance to each landmark to the found coordinate

	Average
7	0.012767
9	0.013241
2	0.016047
8	0.016807
4	0.016890
5	0.019915
3	0.022616
1	0.024326
6	0.028338

stat_landmark_crime.py:
- landmark coordinates near crime coordinates are removed
- finds the averaging center point of landmarks based on each neighborhood's landmark coordinates
- then finds the average distance to each landmark to the found coordinate

	Average
9	0.015499
5	0.020033
6	0.024395
2	0.025446
8	0.025576
1	0.028787

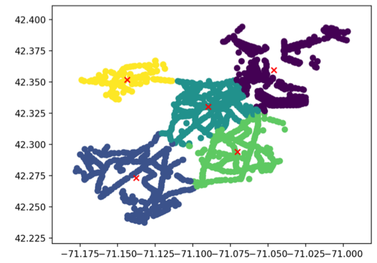
stat_landmark_transportation.py:
- finds the averaging center point of landmarks and transportations based on each neighborhood's landmark coordinates and transportation coordinates
- then finds the average distance to each landmarks and transporations to the found coordinates

	Average
7	0.012774
9	0.013241
2	0.016053
4	0.016843
8	0.016845
5	0.019936
3	0.022617
1	0.024326
6	0.028309

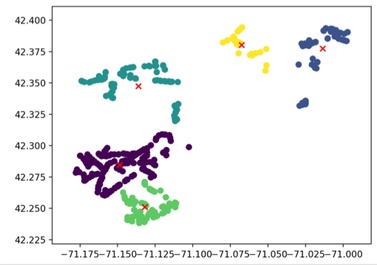
stat_landmark_transportation_crime.py:
- landmark coordinates near crime coordinates are removed
- finds the averaging center point of landmarks and transportations based on each neighborhood's landmark coordinates and transportation coordinates
- then finds the average distance to each landmarks and transporations to the found coordinates

	Average
7	0.013774
9	0.013912
2	0.014489
8	0.017705
4	0.019014
5	0.019202
3	0.020754
1	0.025565
6	0.026709

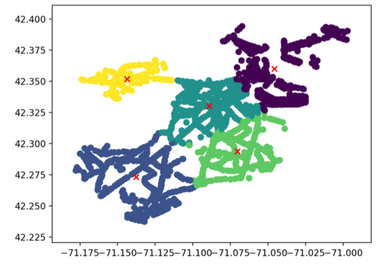
k-means_landmark.py:
- K-means algorithm for finding clusters of landmarks
- locates K coordinates that are centers of the found clusters



k-means_landmark_crime.py:
- K-means algorithm for finding clusters of landmarks
- landmark coordinates close to crime coordinates are removed
- locates K coordinates that are centers of the found clusters



k-means_landmark_transportation.py:
- K-means algorithm for finding clusters of landmarks and transporations (bus&train)
- locates K coordinates that are centers of the found clusters



k-means_landmark_transportation_crime.py:
- K-means algorithm for finding clusters of landmarks and trasnportations (bus&train) where landmark or transporation coordinates close to crime coordinates are removed
- locates K coordinates that are centers of the found clusters