Description of functions

accessUsers

Data must be sorted by owner before passing into this function. (Use excel to sort your csv before converting it into a data frame). Takes in a data frame and counts the number of photos each user has contributed and returns a new data frame. This allows you to see how much each user has contributed to the data set.

Sample output csv.

	numOfPhotos
7630637@	550
90365553@	279
24537422@	209
36386822@	170
31109147@	114

appendUserInfo

Takes in a data frame, retrieves the user specified home location from the Flickr API. Adds the data as a new column and returns a data frame. If the user did not provide any info the location is “none”.
Sample output csv.

	id	accuracy	latitude	longitude	owner	datetaken	title	tags	location
0	5.02E+10	16	51.98573	-176.611	10677383	5/17/2019 13:56	Adak scen	alaska	Corrales, NM,
1	5.02E+10	16	51.91027	-176.591	10677383	5/17/2019 16:39	Adak scen	alaska	Corrales, NM,
2	5.02E+10	16	51.83543	-176.622	10677383	5/16/2019 12:40	Gather no	moss-764(	Corrales, NM,
3	4.95E+10	16	51.90012	-176.621	10677383	5/18/2019 17:17	Rock Sand	adak alask	Corrales, NM,

AverageOfPhotos

It takes in the longitude and latitude coordinates for a geographical bounding box. Makes successive calls to the API and returns an average number of photos for the requested area. It’s hardcoded to make 10 calls. The code can be adjusted to make more calls.

bboxWithCoord

Helper function used with createDataFrame . It takes in longitude and latitude coordinates and makes the Flickr API call. It returns the results from the API Call. It is not designed for you to call directly.

createDataFrame

Creates a new data frame with the information for an API call. Takes in the longitude and latitude coordinates for a bounding box and returns a data frame with the requested info. It relies on information from the constructor and the bounding box coordinates that are passed in. It can be used on its own, but it most of the time it serves as a helper method for recursiveBBox.

dropDuplicates

Takes in a data frame, identifies, and eliminates duplicates, and returns a new data frame. The API often returns duplicate photos and it's good to check for them. This is also useful if geographic bounding boxes overlap.

dropNone

Takes in a data frame that has location info and drops all the rows where the location column = "none". It returns a new data frame.

numOfPhotos

Takes in the coordinates for a bounding box and returns the number of photos in the requested area. It's a helper method for recursiveBBox, but it can also be used on its own.

numOfUsers

Takes in a data frame and returns the number of users. Want to know how many users are in your data set? This tool makes it easy!

Prefill

Takes in a data frame and creates a column for location code. It uses the values of 'v' for visitor and 'l' for local. It looks for words that you identify as being associated with locals and records an 'l' in the location code column. All other entries receive a 'v'. You must specify your words inside the function.

Input csv example

A	B	C	D	E	F	G	H	I	J
	id	accuracy	latitude	longitude	owner	datetaken	title	tags	location
118459	2.71E+10	16	43.09279	-70.7662	100327756	5/21/2016 2:57	#piscataq	piscataqu	Philadelphia, PA, USA
99814	1.47E+10	16	43.91199	-69.9101	100164001	8/10/2014 17:16	20140810_171659		Brunswick, ME, USA
34767	7.1E+09	15	43.62982	-70.5123	10120458(	5/13/2010 3:44	Working h	horses ma	Buxton, Maine, USA
8223	9E+08	16	43.68086	-70.3701	10481380(	7/14/2007 9:33	Riverfront	26jul07de	Westbrook, Maine, United
97527	1.38E+10	16	44.55222	-70.5486	104828224	4/5/2014 8:49	WIN_20140405_0849		Rumford, maine, usa
35176	5.04E+09	16	43.13424	-70.6477	10531415(	9/18/2010 18:02	York Main	york light	Kittery, me
53161	5.97E+09	15	43.65642	-70.2784	10023112(	7/24/2011 20:06	IMG_1560	portland b	Chicago, United States

Changes in the code:

```
for words in locationWords:
    #test for local
    if words == "maine":
        result = "local"
        break
    elif words == "me":
        result = "local"
        break
    #elif words == "yourterm":
        #result = "local"
        #break
    #elif words == "yourterm":
        #result == "local"
        #break
    #elif words == "yourterm":
        #result = "local"
        #break
    #elif words == "yourterm":
        #result = "local"
        #break
```

All the text is converted into lowercase. Your search words should be lowercase as well. There is extra room for additional search terms, just uncomment the code to add them.

Example Output csv

A	B	C	D	E	F	G	H	I	J	K	L	M
	Unnamed	id	accuracy	latitude	longitude	owner	datetaken	title	tags	location	test_loc	location code
0	118459	2.71E+10	16	43.09279	-70.7662	100327756	########	#piscataq	piscataqu	Philadelp	none	v
1	99814	1.47E+10	16	43.91199	-69.9101	100164001	########	20140810_171659		Brunswick	local	l
2	34767	7.1E+09	15	43.62982	-70.5123	10120458(	########	Working h	horses ma	Buxton, M	local	l
3	8223	9E+08	16	43.68086	-70.3701	10481380(	########	Riverfront	26jul07de	Westbroo	local	l
4	97527	1.38E+10	16	44.55222	-70.5486	104828224	########	WIN_20140405_0849		Rumford,	local	l
5	35176	5.04E+09	16	43.13424	-70.6477	10531415(	########	York Main	york light	Kittery, m	local	l
6	53161	5.97E+09	15	43.65642	-70.2784	10023112(	########	IMG_1560	portland b	Chicago, U	none	v

This technique produces a couple of extra columns (B and L) that can easily be deleted in excel. The results are very good, but occasionally this technique will fail to identify a positive case. I usually double check my results manually as well.

recursiveBBox

Takes in the longitude and latitude coordinates for a geographic bounding box and returns a data frame with the requested info. It checks the requested number of photos available for a bounding box, if the number of photos is greater than 4,000, it splits the box into quarters. Once the number of photos is less than 4,000, it pulls the data for the photos. It puts out a data frame that contains all the photos in the requested area. This function creates the initial data frame that can be fed into other functions.

Example output csv

	A	B	C	D	E	F	G	H	I
1		id	accuracy	latitude	longitude	owner	datetaken	title	tags
2	126118	2.97E+10	16	43.6651	-70.2691	1000229€	10/1/2016 0:57	Portland,	instagram
3	126117	2.97E+10	16	43.6651	-70.2691	1000229€	10/1/2016 1:45	View over	instagram
4	126121	2.98E+10	16	43.62306	-70.2078	1000229€	10/2/2016 17:26	Cliffs at P	instagram
5	126120	3E+10	16	43.62306	-70.2078	1000229€	10/2/2016 17:28	Portland H	instagram

retrieveUserInfo

Helper function used with appendUserInfo. It calls flickr.people.getinfo to retrieve location information from a user's profile. It is not designed to be used alone.

userDensity

Takes in a data frame that has a column for location code and condenses photos that have the same owner, latitude, and longitude. It condenses a series of photos from one owner in the same location into one row (or point, on a map) and outputs a data frame. This helps deal with contribution bias that is usually present in data generated from online communities. It is designed for use with point density or kernel density tools.

Example input csv

	A	B	C	D	E	F	G	H	I	J	K	L
1		id	accuracy	latitude	longitude	owner	datetaken	title	tags	location	location code	
2	149214	5.08E+10	16	46.13298	-70.7475	47441022@	12/30/2020 14:42	Ã‰cureu	squirrel Ã	Saint-Geo	v	
3	149350	5.1E+10	16	43.65303	-70.2687	55634677@	12/30/2020 7:14	Morning L	congresss	Portland,	l	
4	148854	5.08E+10	16	43.82407	-69.7103	12357841@	12/29/2020 15:30	Flag Wave	maine gec	Brunswick	l	
5	149215	5.08E+10	16	46.13298	-70.7475	47441022@	12/29/2020 14:06	Sizerin fla	sizerinflar	Saint-Geo	v	
6	149289	5.08E+10	16	46.07719	-70.6404	54788905@	12/29/2020 9:40	MÃ©sang	mÃ©sang	QuÃ©bec	v	

Example output csv

	A	B	C	D	E	F
1		latitude	longitude	owner	location c	photo count
2	('test', 44.	44.12773	-68.8749	test	l	0
3	('47441022	46.13298	-70.7475	47441022@	v	18
4	('5563467	43.65303	-70.2687	55634677@	l	1
5	('1235784	43.82407	-69.7103	12357841@	l	1
6	('5478890	46.07719	-70.6404	54788905@	v	1

userDensityRaw

It does the same thing as userDensity, but it doesn't need the location code column.

Example output csv.

	A	B	C	D	E
1		latitude	longitude	owner	photo count
2	('146757535	43.41	-70.4906	146757535	94
3	('170252897	44.00208	-70.5734	170252897	92
4	('156772709	43.09102	-70.7555	156772709	84
5	('40712146@	44.42892	-69.0079	40712146@	77
6	('8764562@I	43.89732	-69.9808	8764562@	64
7	('34101606@	43.0886	-70.8263	34101606@	58
8	('40712146@	44.09435	-69.8406	40712146@	57
9	('43565937@	43.90741	-69.6165	43565937@	55