

Scraping and Mapping Instagram Photos with Python, R and Leaflet

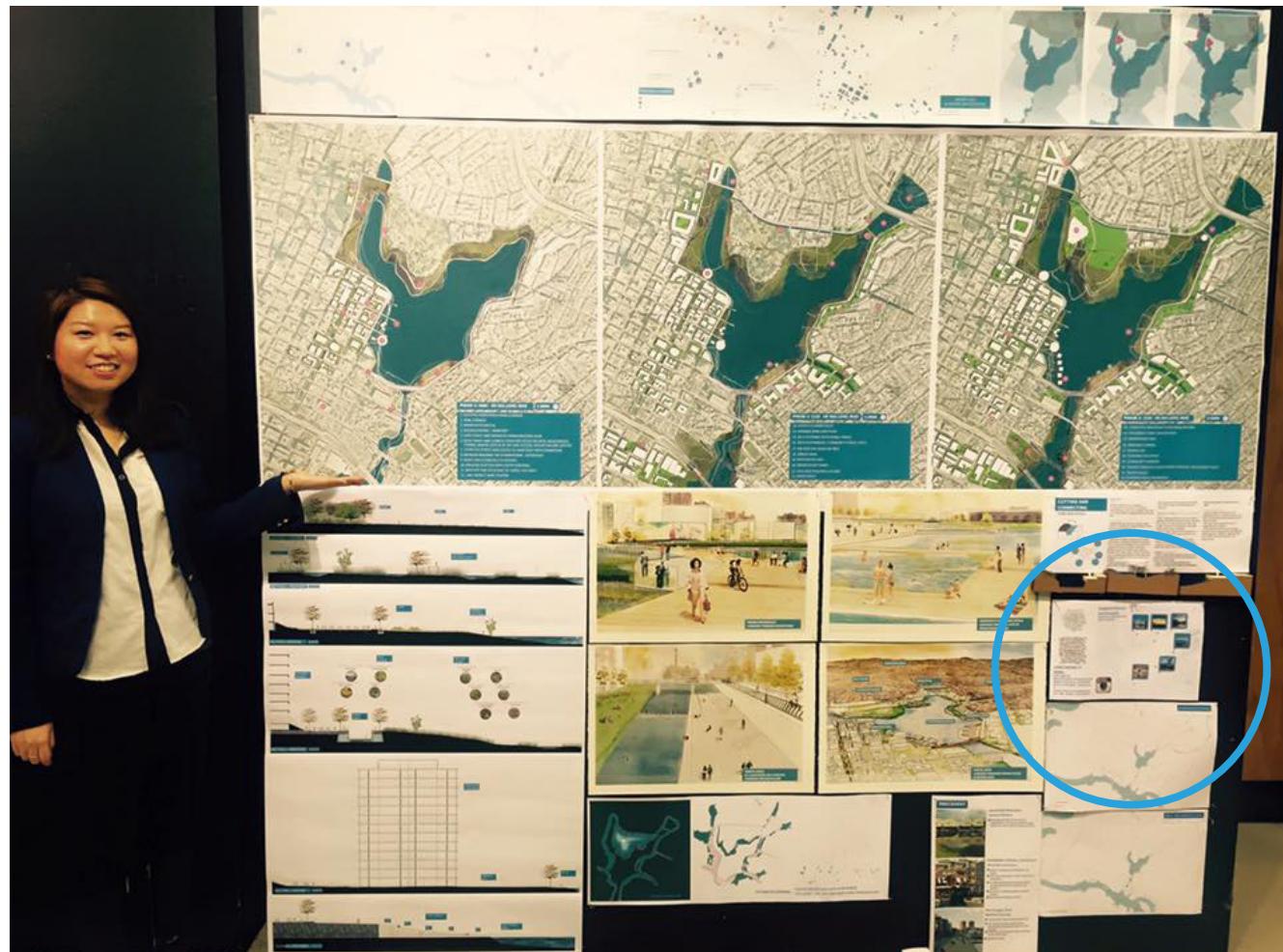
Chelsea Zhou

Context



Design Review: Sea Level Rise Adaptation Proposal for Lake Merritt

How PS239T has helped me form my design strategy...

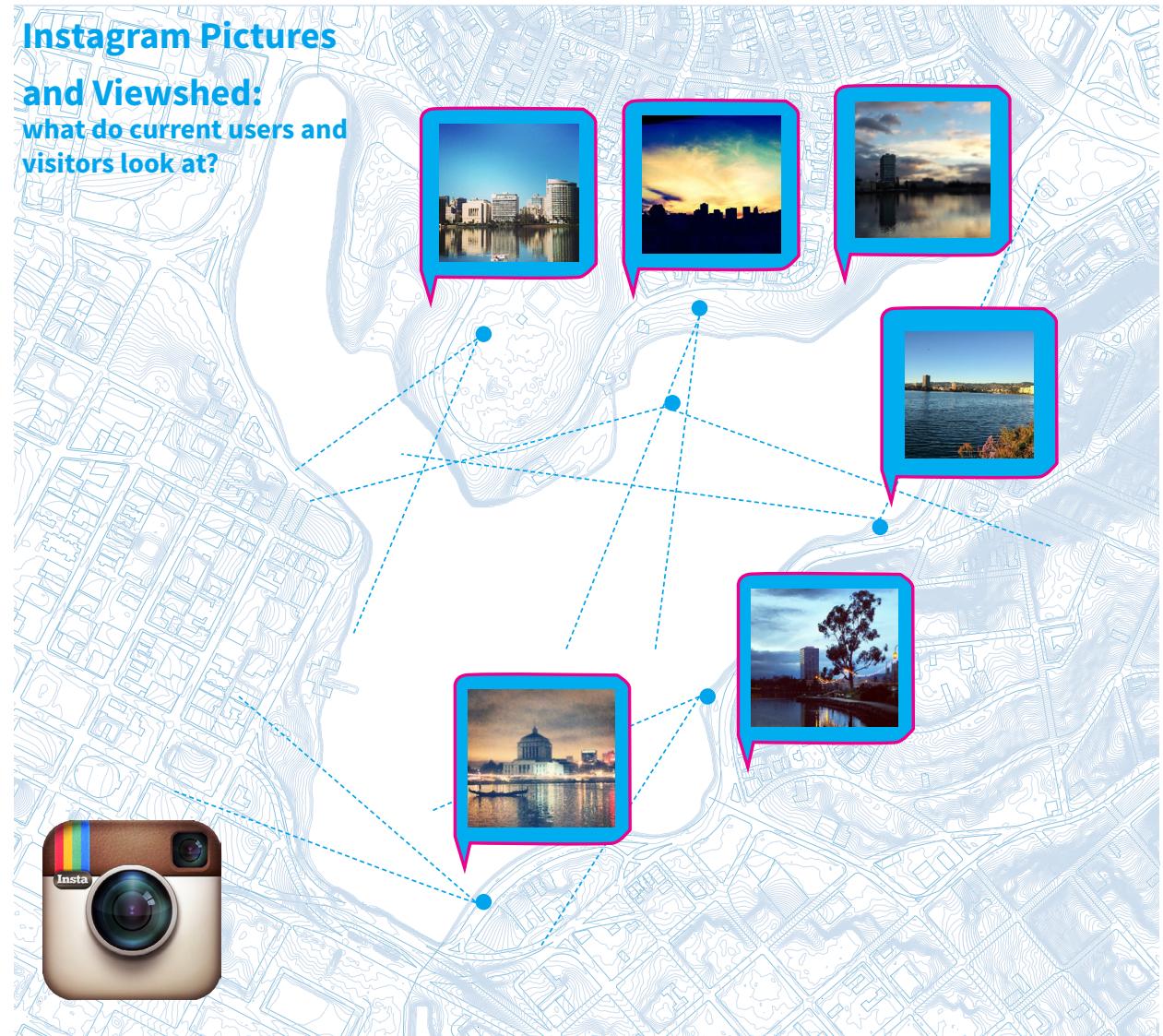


Socioeconomic focus makes my design unique in an ecological design studio :)

project
dog also
help
new two
said
oakland
garden group
inform s includ station center
another member part s hear improv water
see tree o mani resu sayman offic
lakeshore bay tund now sitelive housway need
light open time accord public along event
busi build car around start prop last east
official block park year
street just walk meet found crime local fire
boat name saturday tuesday use
lakesid walk take day measur
people police develop grand one
police develop grand one
at district plan make
million work will
community place

environmental day unit construction benefit
members percent housing business days
students francisco residential busi days
trustive items buildings president
story historic san city million bags
views people put
st merritt project projects
glass hall local director
group high lot public
district area years
tower site urban water found
office community building rise
silver development proposed open
future service school downtown
piedmont cathedral mural news zoning
executive meeting property board making
heard design oakland's

LAKE MERRITT
NOW...
DESCRIBED BY
MOST FREQUENT WORDROOTS AND
WORDS USED IN ALL NEWSPAPER
ARTICLES BETWEEN 1980 and now
&
SAMPLE PUBLIC INSTAGRAM PHOTOS THAT TAGGED 'LAKE MERRITT'



Process

1. Install python-Instagram
2. on instagram developer site, register a new client to get client id and client secret

The screenshot shows a web interface for managing Instagram clients. At the top, there's a header with the client name "Lake Merritt". Below the header are three buttons: "DELETE", "RESET SECRET", and "EDIT". Underneath the header, there's a section titled "CLIENT INFO" which contains two entries: "CLIENT ID" followed by the value "b5d78f65c2b94e74ac6363d8f2f7bcdb" and "CLIENT SECRET" followed by the value "6e00cda85eca47b281af4aa88275c5de".

3. web scrape geolocation (long and lat) with python

```
#THESE ARE PUBLIC PHOTOS TAGGED WITH 'LAKE_MERRITT'

from instagram.client import InstagramAPI
from random import randint
import sys
import time
from collections import OrderedDict

client_id = 'b5d78f65c2b94e74ac6363d8f2f7bcdb'
client_secret = '6e00cda85eca47b281af4aa88275c5de'
access_token = '19594423.1677ed0.d298d661a5dd4587becb8a83b3cca787'
client_ip = '73.189.135.127'

#client_id=client_id, client_secret=client_secret, redirect_uri=redirect_uri

api = InstagramAPI(client_id=client_id, client_secret=client_secret, client_ip=client_ip, access_token=access_token)

media_with_location=[]
all_media_ids = []
```

Process

5. Output to a csv. file

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	installID	lat	long	URL										
2	1022203687	37.7825803	-122.21532	https://scontent.cdninstagram.com/hphotos-xaf1/t50.2886-16/11722892_908448552564991_636535954_n.mp4										
3	9206254568	37.8044143	-122.2515	https://scontent.cdninstagram.com/hphotos-xpa1/t51.2885-15/e15/924756_412596178887551_2004166462_n.jpg										
4	8772403904	37.8073724	-122.25431	https://scontent.cdninstagram.com/hphotos-xpa1/t51.2885-15/e15/927738_751654308260433_880924362_n.jpg										
5	86449509780	37.8068879	-122.25881	https://scontent.cdninstagram.com/hphotos-xpa1/t51.2885-15/e15/924055_1523219964607110_1900805353_n.jpg										
6	8178149801	37.7991161	-122.25891	https://scontent.cdninstagram.com/hphotos-xpa1/t51.2885-15/e15/10522790_448992538577149_1778692219_n.jpg										
7	7428626330	37.8055368	-122.2543	https://scontent.cdninstagram.com/hphotos-xpa1/t50.2885-15/e15/10472465_1435626623377172_669011699_n.mp4										
8	7427847224	37.8055363	-122.2543	https://scontent.cdninstagram.com/hphotos-ash1/t51.2885-15/e15/10467700_1413318752289124_1571403272_n.jpg										
9	7427521174	37.8055363	-122.2543	https://scontent.cdninstagram.com/hphotos-xpa1/t51.2885-15/e15/914364_1433865943542357_581261792_n.jpg										
10	7427304116	37.8055363	-122.2543	https://scontent.cdninstagram.com/hphotos-xaf1/t51.2885-15/e15/10375594_72422190950516_2015127789_n.jpg										
11	742720612	37.8055363	-122.2543	https://scontent.cdninstagram.com/hphotos-xpa1/t51.2885-15/e15/10471849_1419596931655546_130645601_n.jpg										
12	7427069147	37.8055363	-122.2543	https://scontent.cdninstagram.com/hphotos-xaf1/t51.2885-15/e15/927075_29339544168352_1331455806_n.jpg										

6. Plot with R using ggplot



- 1136520041533093719_36388148
- 1136526391819628017_261869346
- 113654700051269651_2236291503
- 1136574131270327446_1648069358
- 1136586489071684171_1821747065
- 1136615185645123569_36388148
- 1136629651261863832_2033817284
- 1136630087502324400_47431591
- 1136664762613747003_144539850
- 1136665935009548233_1556740512
- 1136686432120530316_481965824
- 1136740915059391794_11209382
- 1136774877628166401_1536883
- 1137027798000428440_194350984
- 1137060270479437018_806402234
- 1137135845629619471_829906914
- 1137143853083169722_2267305675

I'm somewhat happy with the result, but...

Next steps

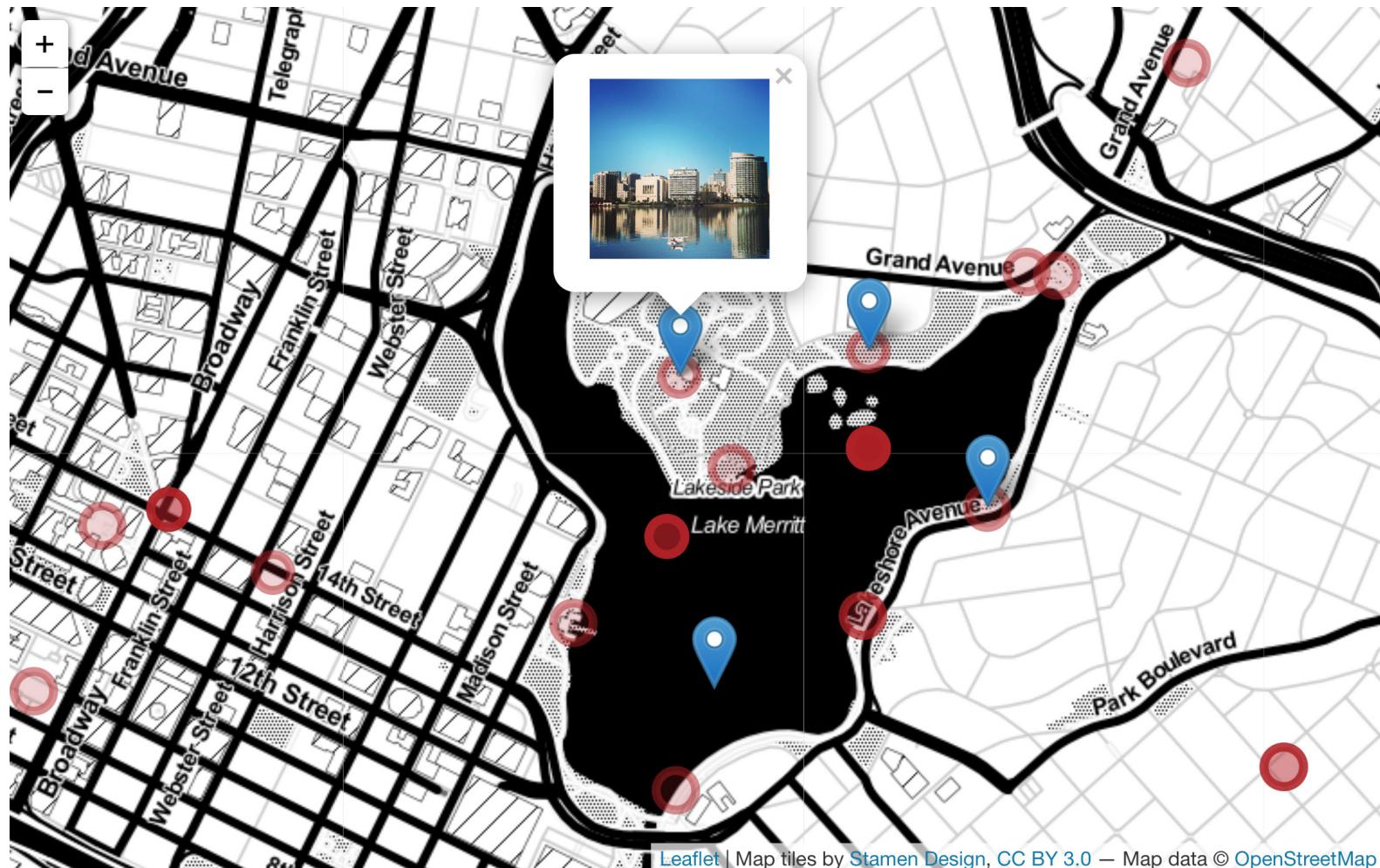
7. Make the map interactive with Leaflet



8. Explore how to display images in popups/markers
9. Share it with the rest of the world

Updates

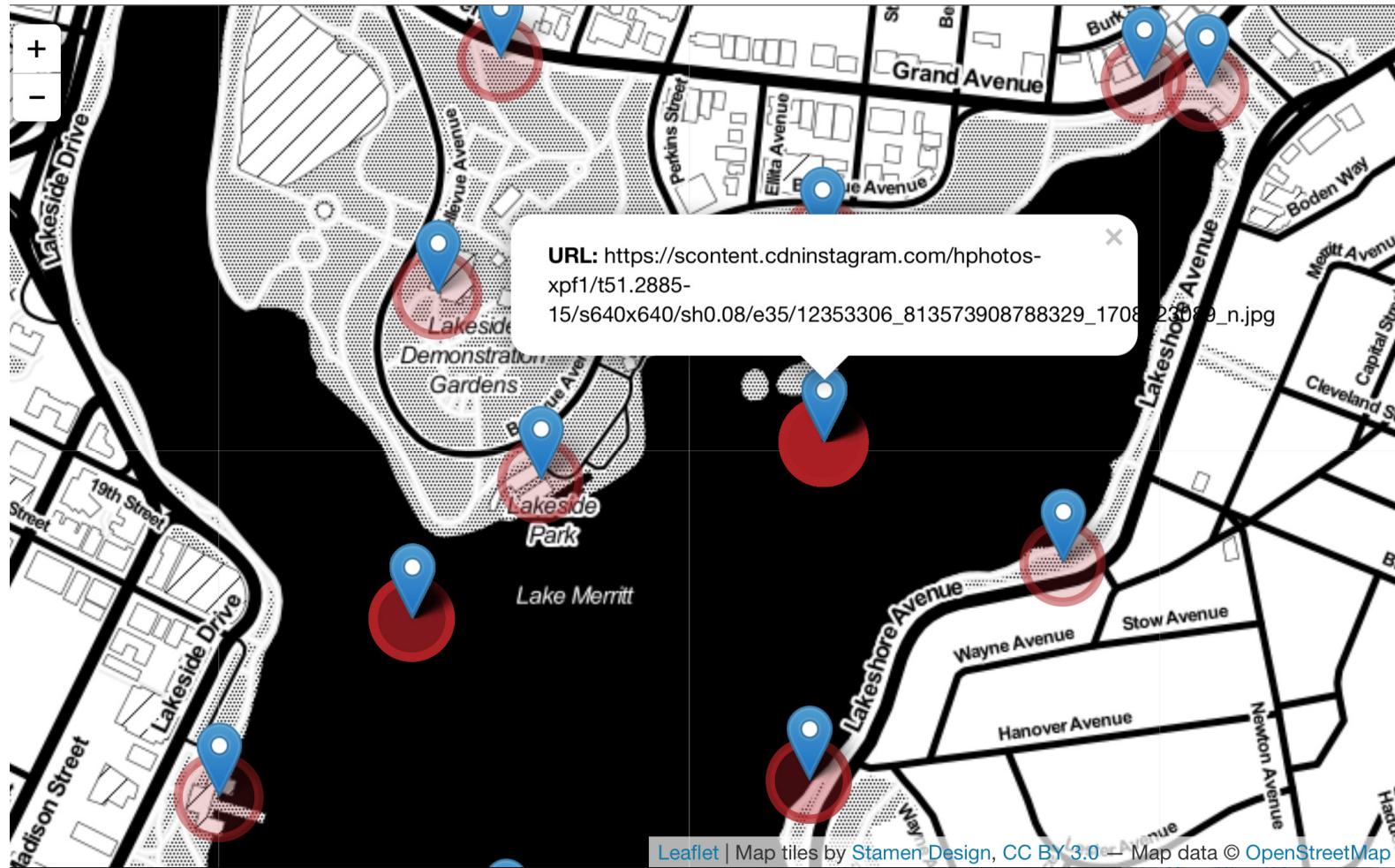
zoom in view



8. Explore how to AUTOMATE embedding images in popups/markers
9. Share it with the rest of the world

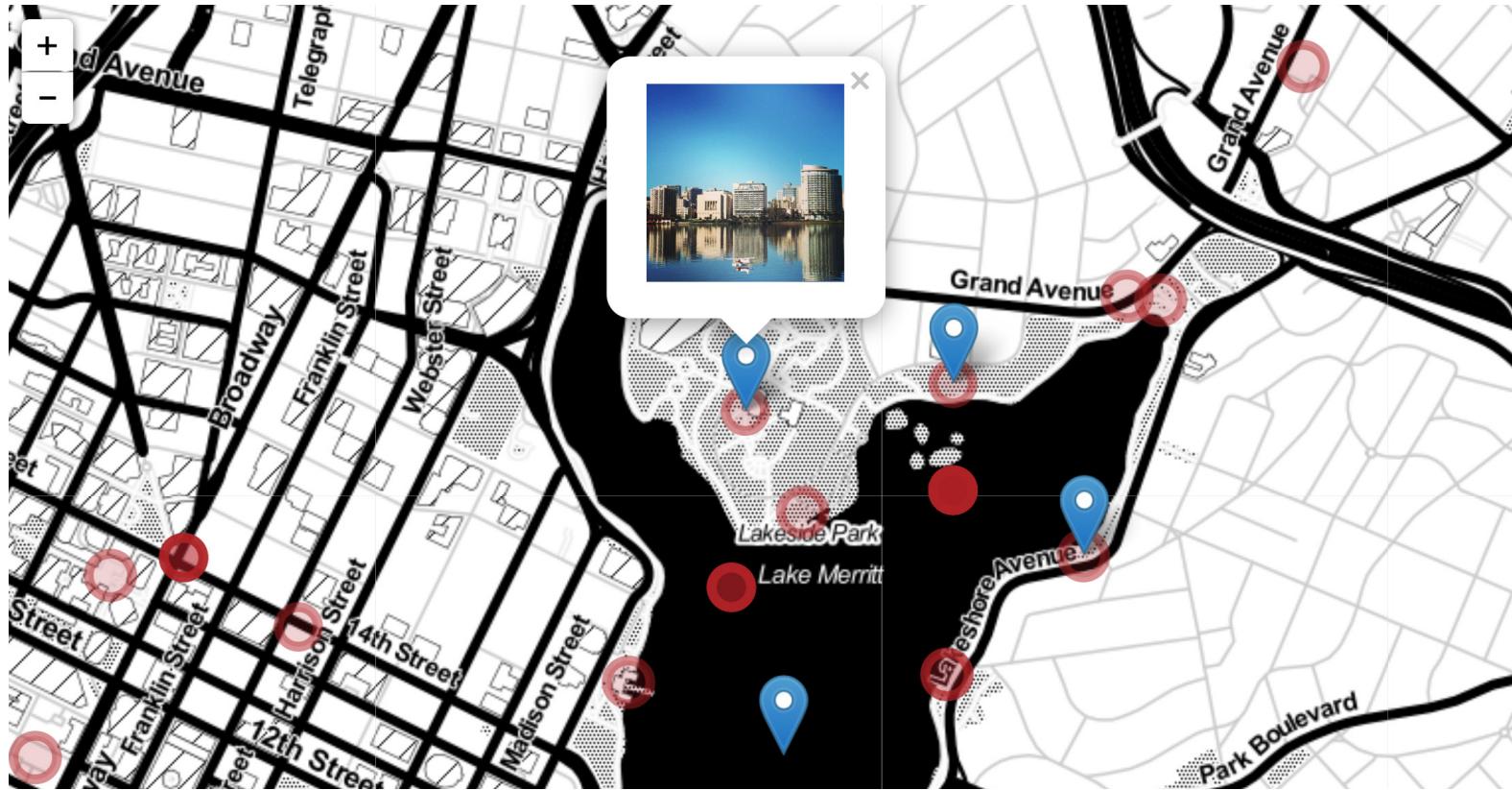
Next steps

zoom in view



8. Explore how to embed images in popups/markers
9. Share it with the rest of the world

Updates



```
, lat = dat$lat, radius = 40, color = firebrick ) %>%  
t=dat$lat[2], popup = '<img src="https://scontent.cdninstagram.com/  
t=dat$lat[3], popup = '<img src="https://scontent.cdninstagram.com/  
t=dat$lat[4], popup = '<img src="https://scontent.cdninstagram.com/
```

8. Explore how to AUTOMATE embedding images in popups/markers
9. Share it with the rest of the world