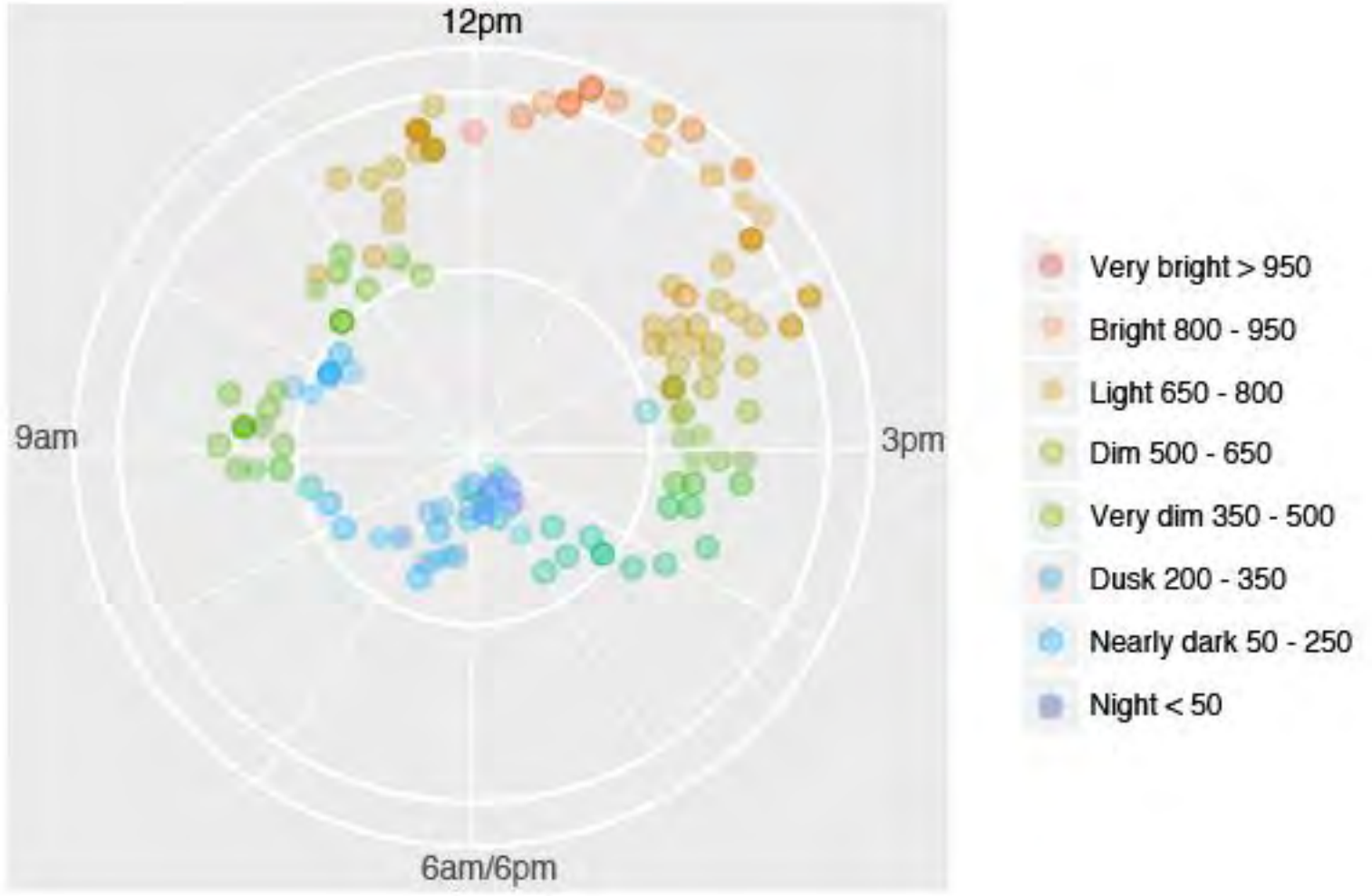
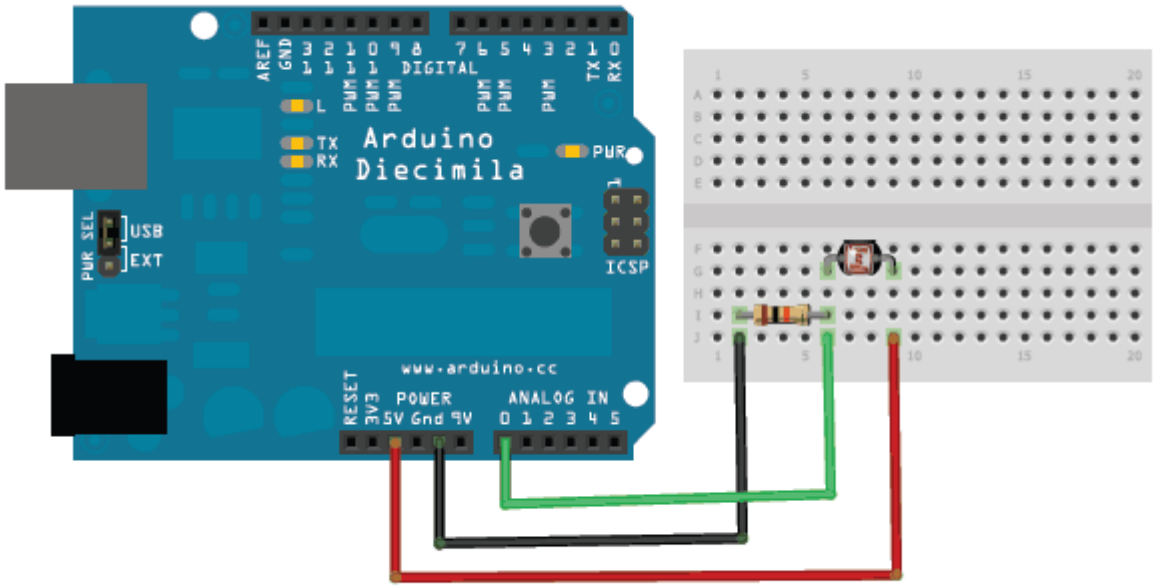
Data Structures Fall 2016:

document database (MongoDB NoSQL) / tabular database (AWS PostgreSQL)



Project 1 – AWS database (PostgreSQL, RDS), photocell sensor, Arduino board:

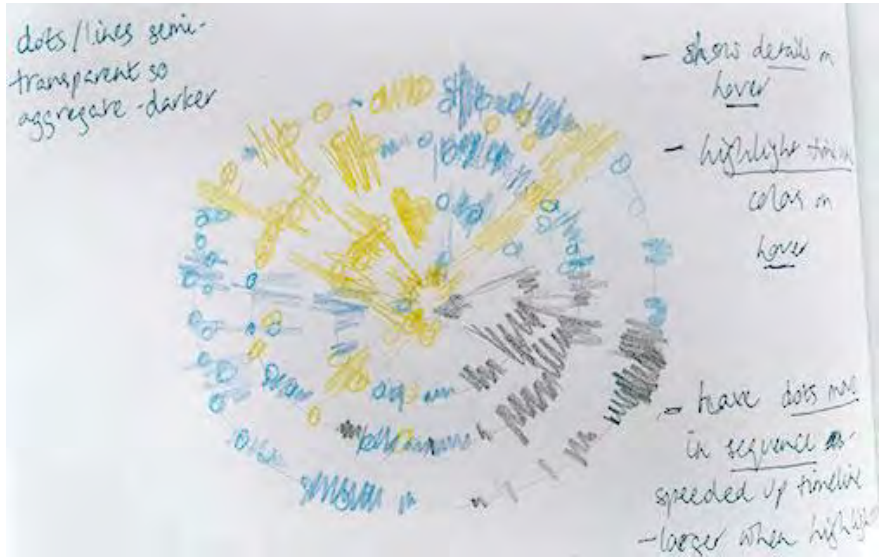
- Collect light data from windowsills:
  - … 12 hours & 3 days & 2 windows
  - … 2 Arduino boards with 12000 delay
  - … Using sublime & terminal, stream data into AWS RDS
- AWS RDS table, 3 columns:
  - … Window number
  - … Light reading
  - … Time
- SQL query on each day / each window. Light readings grouped by brightness and ordered by time to plot light level for the 2 locations
- Visualization based on clock

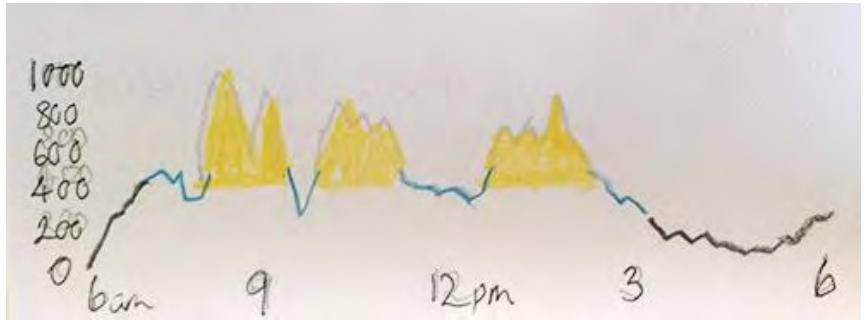


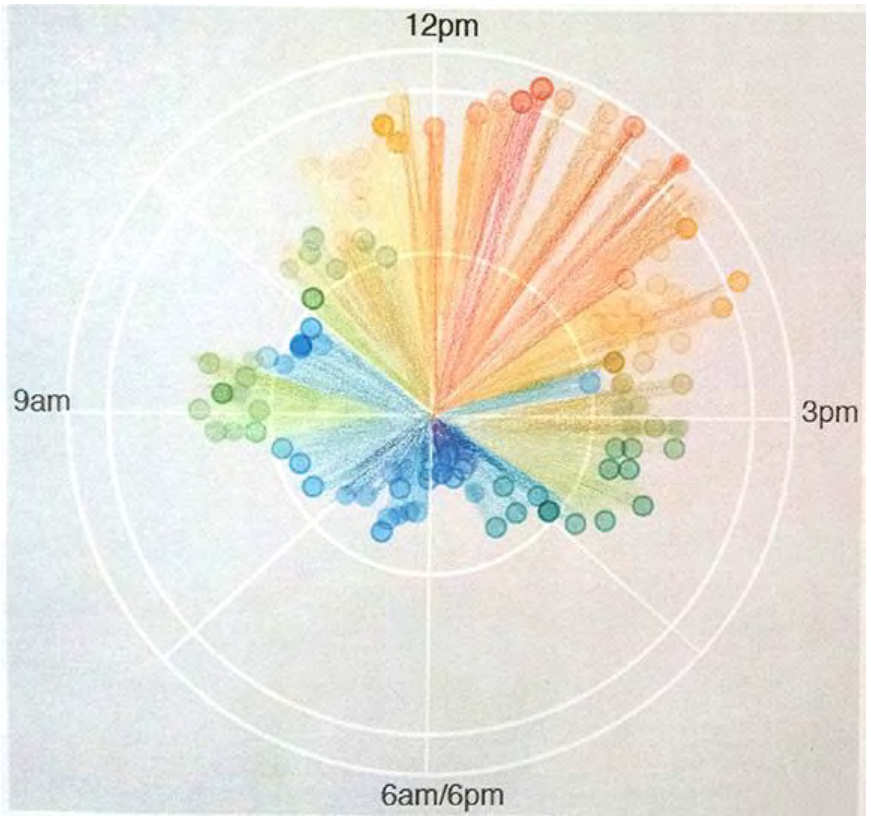
Query AWS db

SQL queries – to group data, change time to EMT, select 6.30am – 6.30pm, simplify time to hh:mm, and order by time

```
///FOR CIRCULAR CHARTS: readings grouped into brightness for Dec 7 - these will each be color coded to read differently on circular chart
var queryNight07 = "SELECT windowNo, reading, EXTRACT(HOUR from dateCreated) as hr, EXTRACT(MINUTE from dateCreated) as min FROM lightP AS nt WHERE dateCreated >= '2016-12-07 11:30:00' AND dateCreated <'2016-12-07 23:30:00' AND reading < 50 ORDER BY dateCreated;";
var queryNearlyDark07 = "SELECT windowNo, reading, EXTRACT(HOUR from dateCreated) as hr, EXTRACT(MINUTE from dateCreated) as min FROM lightP AS nd WHERE dateCreated >= '2016-12-07 11:30:00' AND dateCreated <'2016-12-07 23:30:00' AND reading > 50 AND reading < 200 ORDER BY dateCreated;";
var queryDusk07 = "SELECT windowNo, reading, EXTRACT(HOUR from dateCreated) as hr, EXTRACT(MINUTE from dateCreated) as min FROM lightP AS dk WHERE dateCreated >= '2016-12-07 11:30:00' AND dateCreated <'2016-12-07 23:30:00' AND reading > 200  AND reading < 350 ORDER BY dateCreated;";
var queryVeryDim07 = "SELECT windowNo, reading, EXTRACT(HOUR from dateCreated) as hr, EXTRACT(MINUTE from dateCreated) as min FROM lightP AS vd WHERE dateCreated >= '2016-12-07 11:30:00' AND dateCreated <'2016-12-07 23:30:00' AND reading > 350 AND reading < 500 ORDER BY dateCreated;";
var queryDim07 = "SELECT windowNo, reading, EXTRACT(HOUR from dateCreated) as hr, EXTRACT(MINUTE from dateCreated) as min FROM lightP AS vb WHERE dateCreated >= '2016-12-07 11:30:00' AND dateCreated <'2016-12-07 23:30:00' AND reading > 500 AND reading < 650  ORDER BY dateCreated;";
var queryLight07 = "SELECT windowNo, reading, EXTRACT(HOUR from dateCreated) as hr, EXTRACT(MINUTE from dateCreated) as min FROM lightP AS lr WHERE dateCreated >= '2016-12-07 11:30:00' AND dateCreated <'2016-12-07 23:30:00' AND reading > 650 AND reading < 800 ORDER BY dateCreated;";
var queryBright07 = "SELECT windowNo, reading, EXTRACT(HOUR from dateCreated) as hr, EXTRACT(MINUTE from dateCreated) as min FROM lightP AS bt WHERE dateCreated >= '2016-12-07 11:30:00' AND dateCreated <'2016-12-07 23:30:00' AND reading > 800 AND reading < 950 ORDER BY dateCreated;";
var queryVeryBright07 = "SELECT windowNo, reading, EXTRACT(HOUR from dateCreated) as hr, EXTRACT(MINUTE from dateCreated) as min FROM lightP AS vb WHERE dateCreated >= '2016-12-07 11:30:00' AND dateCreated <'2016-12-07 23:30:00' AND reading >= 950 ORDER BY dateCreated;";
```





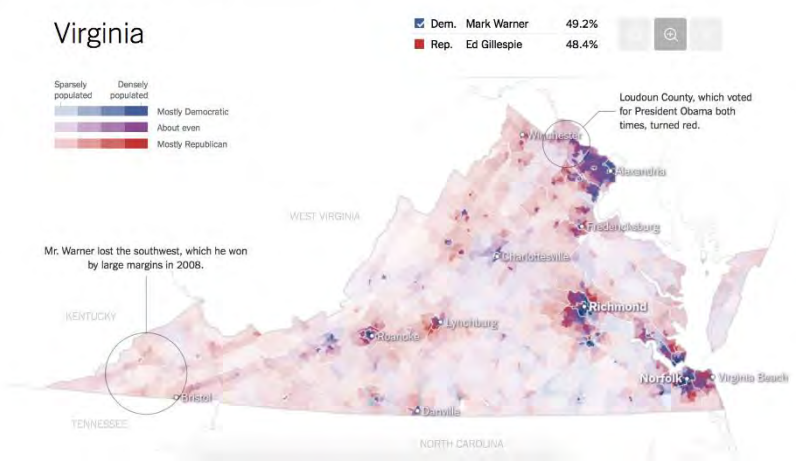


Mockup in r and hand-drawing – not actual data
https://class10-churc.c9users.io/

Next Steps…….
Plot data in r…….

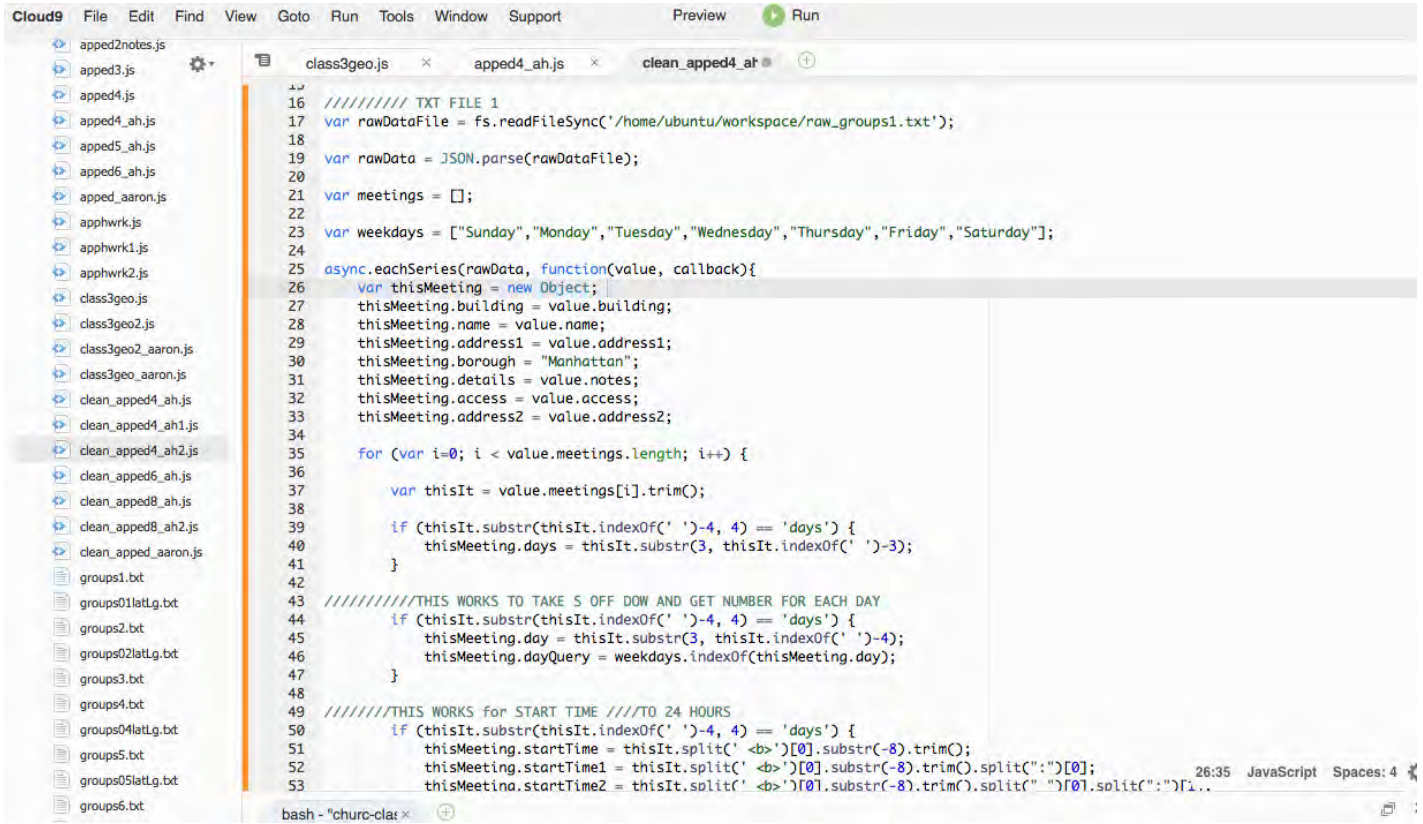
The Most Detailed Maps You'll See From the Midterm Elections*

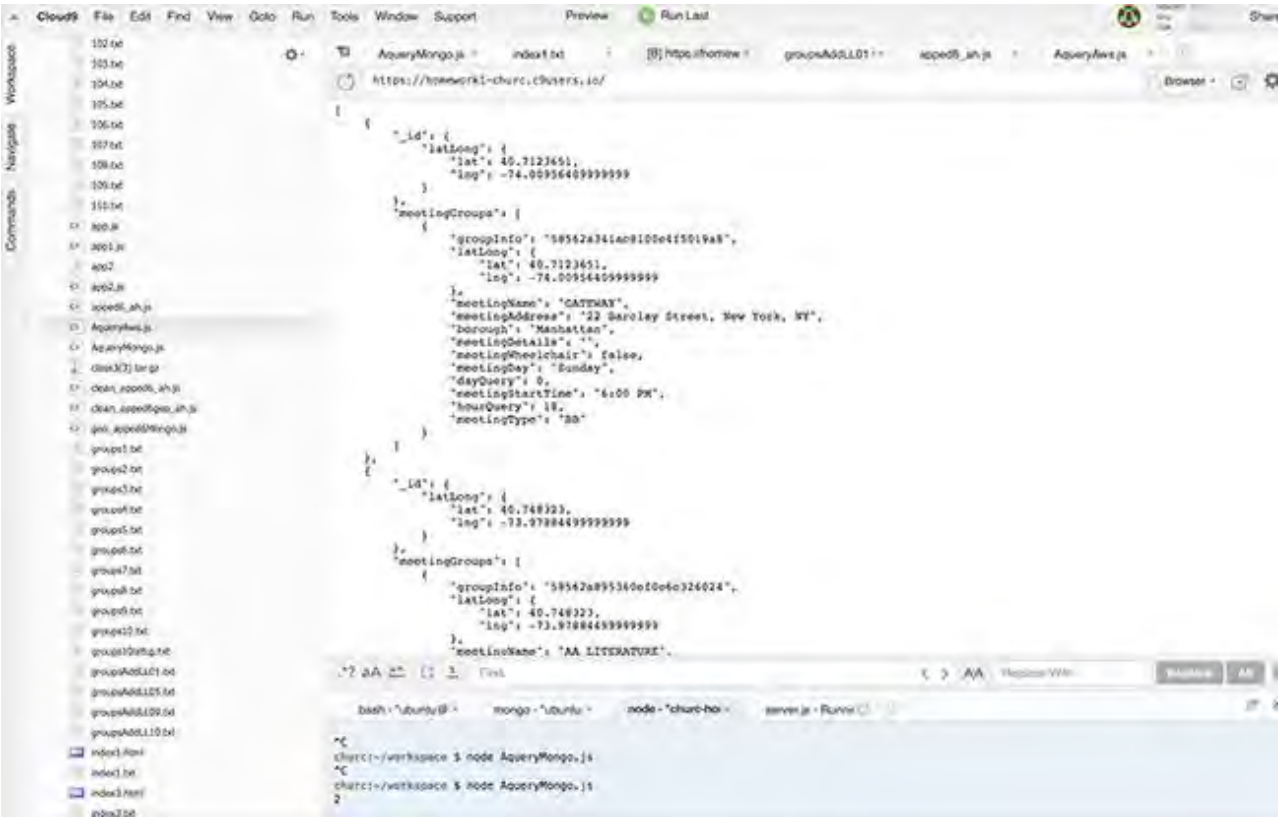
By Amanda Cox, Mike Bostock, Derek Watkins, and Shan Carter Nov. 6, 2014 04:03 PM

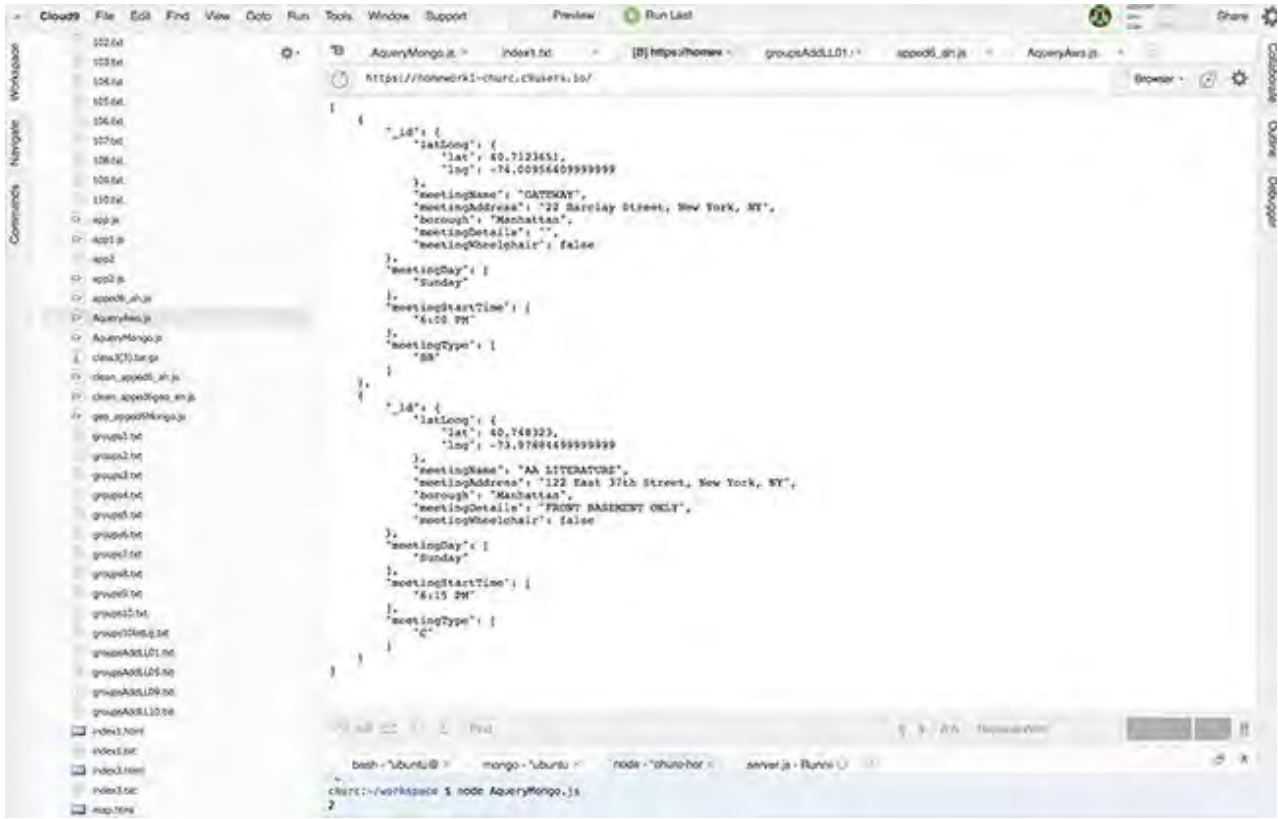


Project 2 – NoSQL, MongoDB

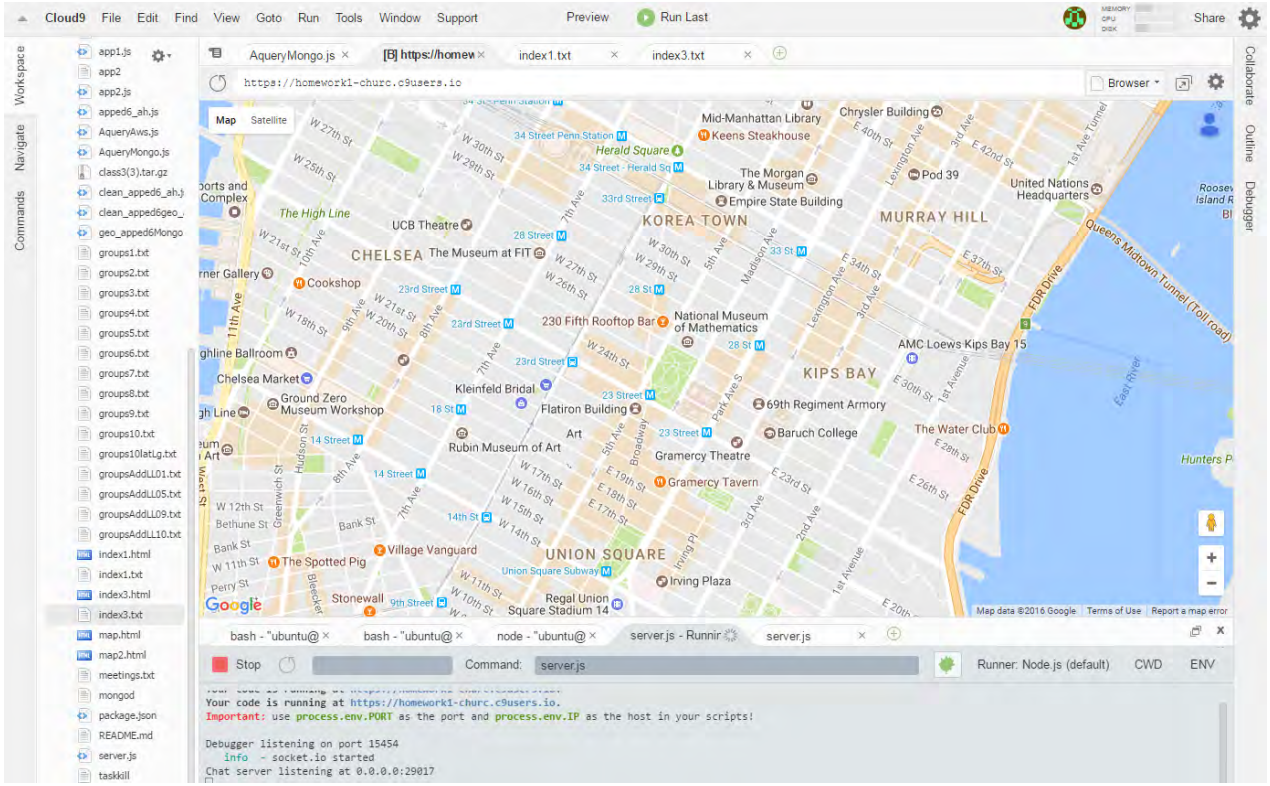
Scrape data from website
clean, store in MongoDB, query output
to Google maps API, filter by location / time




Cloud9 File Edit Find View Goto Run Tools Window Support Preview Run Last
AqueryMongo.js index1.txt [8] https://homew groupsAddLL01.r appe6_ah.js AqueryAws.js
https://homework1-churc.c9users.io/
[
{
"_id": {
"latLong": {
"lat": 40.7123651,
"lng": -74.00956409999999
}
},
"meetingGroups": [
{
"groupInfo": "58562a341ac8100e4f5019a8",
"latLong": {
"lat": 40.7123651,
"lng": -74.00956409999999
},
"meetingName": "GATEWAY",
"meetingAddress": "22 Barclay Street, New York, NY",
"borough": "Manhattan",
"meetingDetails": "",
"meetingWheelchair": false,
"meetingDay": "Sunday",
"dayQuery": 0,
"meetingStartTime": "6:00 PM",
"hourQuery": 18,
"meetingType": "BB"
}
]
},
{
"_id": {
"latLong": {
"lat": 40.748323,
"lng": -73.97884499999999
}
},
"meetingGroups": [
{
"groupInfo": "58562a895360e10e4c326024",
"latLong": {
"lat": 40.748323,
"lng": -73.97884499999999
},
"meetingName": "AA LITERATURE",
bash - "ubuntu@ mongo - "ubuntu node - "churn-hor server.js - Runni
^C
churc:~/workspace $ node AqueryMongo.js
^C
churc:~/workspace $ node AqueryMongo.js
2


Cloud9 File Edit Find View Goto Run Tools Window Support Preview Run Last
AqueryMongo.js index1.txt [8] https://homew groupsAddLL01.r appe6_ah.js AqueryAws.js
https://homework1-churc.c9users.io/
[
{
"_id": {
"latLong": {
"lat": 40.7123651,
"lng": -74.00956409999999
},
"meetingName": "GATEWAY",
"meetingAddress": "22 Barclay Street, New York, NY",
"borough": "Manhattan",
"meetingDetails": "",
"meetingWheelchair": false
},
"meetingDay": [
"Sunday"
],
"meetingStartTime": [
"6:00 PM"
],
"meetingType": [
"BB"
]
},
{
"_id": {
"latLong": {
"lat": 40.748323,
"lng": -73.97884499999999
},
"meetingName": "AA LITERATURE",
"meetingAddress": "122 East 37th Street, New York, NY",
"borough": "Manhattan",
"meetingDetails": "FRONT BASEMENT ONLY",
"meetingWheelchair": false
},
"meetingDay": [
"Sunday"
],
"meetingStartTime": [
"6:15 PM"
],
"meetingType": [
"C"
]
}
]
bash - "ubuntu@ mongo - "ubuntu node - "churn-hor server.js - Runni
churc:~/workspace $ node AqueryMongo.js
2

Had a lot of help
Spent hours on coding
https://homework1-churc.c9users.io/



What parts went well for you?

< Learnt a huge amount
< Arduino boards
< Node
< SQL

Most challenging:

> Javascript
(! {} [] , ; “” .)

If you had to redo all your work in DVIA using databases and a web server module in Node, what would you have done differently?

Would have made maps more interactive to reflect current information, e.g. ACO numbers, and allow for geographic location to pull up details for area the user is located.

Next Steps……..

Continue with Javascript……..