IOWA DOT CONTINUOUS COUNT STATION REFERENCE PAGE

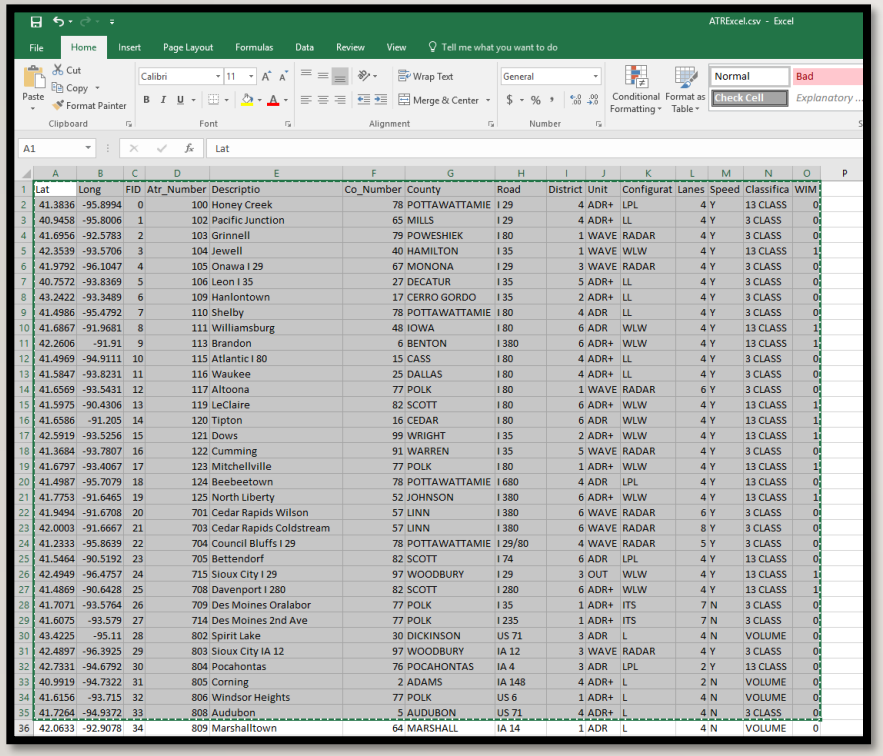
BY: BRANDON GEBER (KLEIN)

LA 558 FINAL PROJECT

CONTINUOUS COUNT STATION DATA

- The Iowa Department of Transportation maintains 173 continuous count stations (Automatic Traffic Recorders) throughout the state of Iowa.
- Many departments use this data, but since there is not a central inventory location, the automatic traffic recorder location data is often out-of-date as units go under repair or are moved.
- The objective of this project is to create a single location for this inventory data that is readily available, easy to keep up-to-date, and will provide staff with resources to evaluate potential anomalies, such as traffic congestion or road construction projects.

CONVERTING DATA TO A WEB-BASED GEOGRAPHIC FILE



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Lat	Long	FID	Atr_Number	Descriptio	Co_Number	County	Road	District	Unit	Configurat	Lanes	Speed	Classifica	WIM	
2	41.3836	-95.8994	0	100	Honey Creek	78	POTTAWATTAMIE	I 29	4	ADR+	LPL	4	Y	13 CLASS	0	
3	40.9458	-95.8006	1	102	Pacific Junction	65	MILLS	I 29	4	ADR+	LL	4	Y	3 CLASS	0	
4	41.6956	-92.5783	2	103	Grinnell	79	POWESHIEK	I 80	1	WAVE	RADAR	4	Y	3 CLASS	0	
5	42.3539	-93.5706	3	104	Jewell	40	HAMILTON	I 35	1	WAVE	WLW	4	Y	13 CLASS	1	
6	41.9792	-96.1047	4	105	Onawa I 29	67	MONONA	I 29	3	WAVE	RADAR	4	Y	3 CLASS	0	
7	40.7572	-93.8369	5	106	Leon I 35	27	DECATUR	I 35	5	ADR+	LL	4	Y	3 CLASS	0	
8	43.2422	-93.3489	6	109	Hanlontown	17	CERRO GORDO	I 35	2	ADR+	LL	4	Y	3 CLASS	0	
9	41.4986	-95.4792	7	110	Shelby	78	POTTAWATTAMIE	I 80	4	ADR	LL	4	Y	3 CLASS	0	
10	41.6867	-91.9681	8	111	Williamsburg	48	IOWA	I 80	6	ADR	WLW	4	Y	13 CLASS	1	
11	42.2606	-91.91	9	113	Brandon	6	BENTON	I 380	6	ADR+	WLW	4	Y	13 CLASS	1	
12	41.4969	-94.9111	10	115	Atlantic I 80	15	CASS	I 80	4	ADR+	LL	4	Y	3 CLASS	0	
13	41.5847	-93.8231	11	116	Waukee	25	DALLAS	I 80	4	ADR+	LL	4	Y	3 CLASS	0	
14	41.6569	-93.5431	12	117	Altoona	77	POLK	I 80	1	WAVE	RADAR	6	Y	3 CLASS	0	
15	41.5975	-90.4306	13	119	LeClaire	82	SCOTT	I 80	6	ADR+	WLW	4	Y	13 CLASS	1	
16	41.6586	-91.205	14	120	Tipton	16	CEDAR	I 80	6	ADR	WLW	4	Y	13 CLASS	1	
17	42.5919	-93.5256	15	121	Dows	99	WRIGHT	I 35	2	ADR+	WLW	4	Y	13 CLASS	1	
18	41.3684	-93.7807	16	122	Cumming	91	WARREN	I 35	5	WAVE	RADAR	4	Y	3 CLASS	0	
19	41.6797	-93.4067	17	123	Mitchellville	77	POLK	I 80	1	ADR+	WLW	4	Y	13 CLASS	1	
20	41.4987	-95.7079	18	124	Beebeetown	78	POTTAWATTAMIE	I 680	4	ADR	LPL	4	Y	13 CLASS	0	
21	41.7753	-91.6465	19	125	North Liberty	52	JOHNSON	I 380	6	ADR+	WLW	4	Y	13 CLASS	1	
22	41.9494	-91.6708	20	701	Cedar Rapids Wilson	57	LINN	I 380	6	WAVE	RADAR	6	Y	3 CLASS	0	
23	42.0003	-91.6667	21	703	Cedar Rapids Coldstream	57	LINN	I 380	6	WAVE	RADAR	8	Y	3 CLASS	0	
24	41.2333	-95.8639	22	704	Council Bluffs I 29	78	POTTAWATTAMIE	I 29/80	4	WAVE	RADAR	5	Y	3 CLASS	0	
25	41.5464	-90.5192	23	705	Bettendorf	82	SCOTT	I 74	6	ADR	LPL	4	Y	13 CLASS	0	
26	42.4949	-96.4757	24	715	Sioux City I 29	97	WOODBURY	I 29	3	OUT	WLW	4	Y	13 CLASS	1	
27	41.4869	-90.6428	25	708	Davenport I 280	82	SCOTT	I 280	6	ADR+	WLW	4	Y	13 CLASS	1	
28	41.7071	-93.5764	26	709	Des Moines Oralabor	77	POLK	I 35	1	ADR+	ITS	7	N	3 CLASS	0	
29	41.6075	-93.579	27	714	Des Moines 2nd Ave	77	POLK	I 235	1	ADR+	ITS	7	N	3 CLASS	0	
30	43.4225	-95.11	28	802	Spirit Lake	30	DICKINSON	US 71	3	ADR	L	4	N	VOLUME	0	
31	42.4897	-96.3925	29	803	Sioux City IA 12	97	WOODBURY	IA 12	3	WAVE	RADAR	4	Y	3 CLASS	0	
32	42.7331	-94.6792	30	804	Pocahontas	76	POCAHONTAS	IA 4	3	ADR	LPL	2	Y	13 CLASS	0	
33	40.9919	-94.7322	31	805	Corning	2	ADAMS	IA 148	4	ADR+	L	2	N	VOLUME	0	
34	41.6156	-93.715	32	806	Windsor Heights	77	POLK	US 6	1	ADR+	L	4	N	VOLUME	0	
35	41.7264	-94.9372	33	808	Audubon	5	AUDUBON	US 71	4	ADR+	L	4	N	3 CLASS	0	
36	42.0633	-92.9078	34	809	Marshalltown	64	MARSHALL	IA 14	1	ADR	L	4	N	VOLUME	0	

- Current inventory data is already kept in an Excel spreadsheet that is shared between the telemetrics team.
- Converting this spreadsheet into a CSV allows the data to be quickly transformed into a GeoJSON file through www.geojson.io
- As data is updated monthly, this file can easily be converted and linked to the webpage.

FILTERING DATA USING JQUERY

```
122
123         var promise = $.getJSON("https://bmklein.github.io/LA558/IowaATR.json");
124             promise.then(function(data) {
125
126
127             var adr = L.geoJson(data, {
128                 filter: function(feature, layer) {
129                     return feature.properties.type == "ADR";
130                 },
131                 pointToLayer: function(feature, latlng) {
132                     return L.marker(latlng, {
133                     icon: adrIcon
134                     });
135                 },
136                 onEachFeature: function (feature, layer) {
137                     htmlText = "<b>Unit ID: </b>" + feature.properties.ID + "<br><b>Location: </b>"  +
138                     layer.bindPopup(htmlText);
139                 }
140             });
141
142
143
144             var adr1 = L.geoJson(data, {
145                 filter: function(feature, layer) {
146                     return feature.properties.type == "ADR+";
147                 },
148                 pointToLayer: function(feature, latlng) {
149                     return L.marker(latlng, {
150                     icon: adrIcon
151                     });
152                 },
153                 onEachFeature: function (feature, layer) {
154                     htmlText = "<b>Unit ID: </b>" + feature.properties.ID + "<br><b>Location: </b>"  +
155                     layer.bindPopup(htmlText);
156                 }
157             });
158
159
160             var wave = L.geoJson(data, {
161                 filter: function(feature, layer) {
```

- Using filter code sequences within jQuery, each interested attribute was filtered out of the new GeoJSON dataset, and setup to be individually represented on the web page.

REPRESENTING DATA USING AWESOME MARKERS

```
49  //Awesome Markers
50          var adrIcon = L.AwesomeMarkers.icon({
51              prefix: 'fa', //bootstrap
52              markerColor: 'green', // see colors above
53              icon: 'fa-car'
54          });
55
56          var waveIcon = L.AwesomeMarkers.icon({
57              prefix: 'fa', //bootstrap
58              markerColor: 'darkpurple', // see colors above
59              icon: 'fa-rss'
60          });
61
62          var wimIcon = L.AwesomeMarkers.icon({
63              prefix: 'fa', //bootstrap
64              markerColor: 'darkblue', // see colors above
65              icon: 'fa-truck'
66          });
67
68          var outIcon =   L.AwesomeMarkers.icon({
69              prefix: 'fa',  //ionicons.com
70              markerColor: 'red',
71              icon: 'fa-times'
72          });
73
```

- Each filtered dataset was matched up with a tailored Awesome Markers marker through Font Awesome.
- Varying colors and icons were used to distinguish between the different recorder attributes.

CONNECTING BUTTONS TO FILTERS

```
412            });
413        });
414
415
416
417
418        </script>
419
420    </div><br><br>
421    <br>
422    <div id="btn-group">
423        <button type="button" id="allUnits" class="btn"><b>All Unit Type</b></button><br>
424        <button type="button" id="ADR" class="btn" style=background-color:#86b300><b>ADR</b></button><br>
425        <button type="button" id="wave" class="btn" style=background-color:#39004d><b>WAVE</b></button><br>
426        <button type="button" id="WIM" class="btn" style=background-color:#0059b3><b>WIM</b></button><br>
427        <button type="button" id="allClass" class="btn" style=margin-top:30px><b>All Class Type</b></button><br>
428        <button type="button" id="volume" class="btn" style=background-color:#b3b3b3><b>Volume</b></button><br>
429        <button type="button" id="3class" class="btn" style=background-color:#595959><b>3-Class</b></button><br>
430        <button type="button" id="13class" class="btn" style=background-color:black><b>13-Class</b></button><br>
431        <button type="button" id="speed" class="btn" style="background-color:orange; margin-top:30px"><b>StopWatch
432        <button type="button" id="out" class="btn" style="background-color:#cc0000; margin-top:30px"><b>Out</b></
433    </div>
434
435    </div>
436
437
438</body>
439
440
441</html>
442
443
444
445
```

- Buttons were added and connected to jQuery functions to geographically filter the points represented on the map, making it easy to discern between points of interest.

CREATING JQUERY FUNCTIONS TO FILTER DATA

```
294         $("#allUnits").click(function() {
295             map.addLayer(adr);
296             map.addLayer(adr1);
297             map.addLayer(wave);
298             map.addLayer(WIM);
299             map.addLayer(out);
300             map.removeLayer(volume);
301             map.removeLayer(Tclass);
302             map.removeLayer(Sclass);
303             map.removeLayer(speed);
304         });
305
306         $("#ADR").click(function() {
307             map.addLayer(adr);
308             map.addLayer(adr1);
309             map.removeLayer(wave);
310             map.removeLayer(out);
311             map.removeLayer(WIM);
312             map.removeLayer(volume);
313             map.removeLayer(Tclass);
314             map.removeLayer(Sclass);
315             map.removeLayer(speed);
316         });
317
318         $("#wave").click(function() {
319             map.removeLayer(adr);
320             map.removeLayer(adr1);
321             map.addLayer(wave);
322             map.removeLayer(out);
```

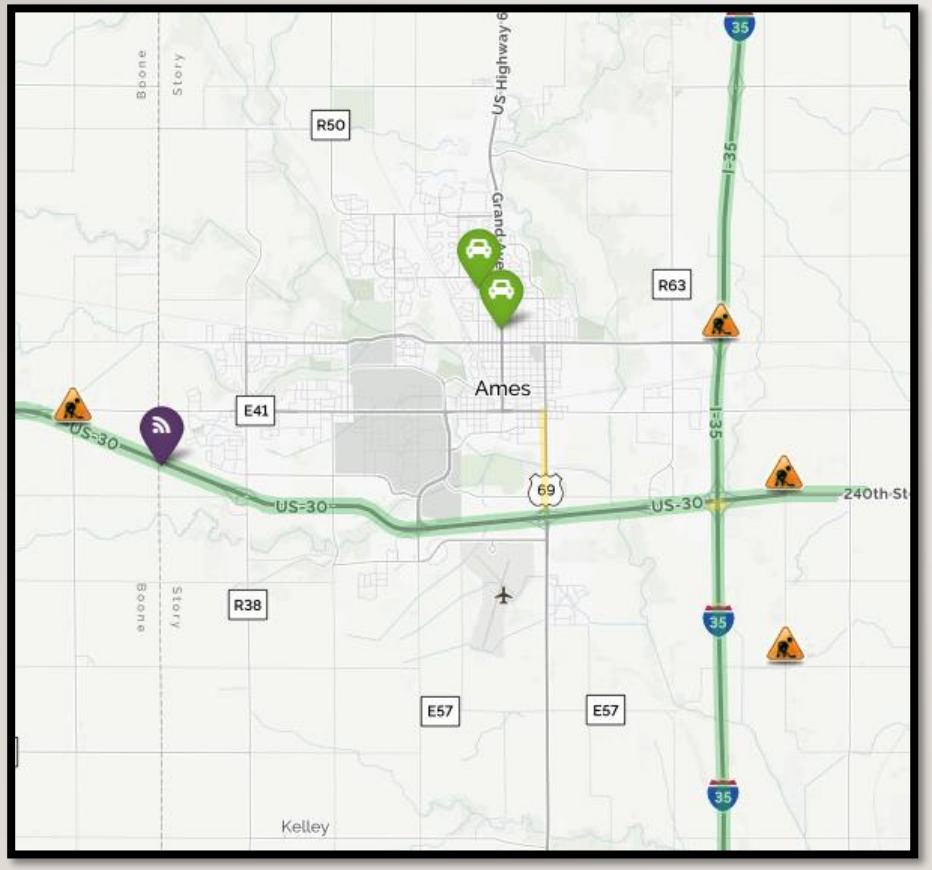
- The jQuery functions connected to the buttons were set up to remove certain filtered points and add the points of interest.

INTRODUCING TRAFFIC FLOW AND INCIDENTS

```
100         //create the maps
101
102
103         var mapLayer = MQ.mapLayer(),
104             map;
105
106             map = L.map('map', {
107                 layers: mapLayer,
108                 center: [ 42.0, -93.0 ],
109                 zoom: 12
110             });
111
112             L.control.layers({
113                 'Map': mapLayer,
114                 'Satellite': MQ.satelliteLayer(),
115                 'Dark': MQ.darkLayer(),
116                 'Light': MQ.lightLayer()
117             }, {
118                 'Traffic Flow': MQ.trafficLayer({layers: ['flow']}),
119                 'Traffic Incidents': MQ.trafficLayer({layers: ['incidents']})
120             }).addTo(map);
121
```

- To aid in issue detection, both traffic flow and traffic incidents data were collected and represented on the map by connecting to the MapQuest API through a free user key.

FINAL REFERENCE PAGE



- The final product results in a web map that displays recorder attributes and locations.
- When activated, the user can visualize traffic flow and incident reports, with traffic flow data becoming more detailed the further the user zooms in.