Portfolio

Daniel Shanker

1 Khan Academy

From January through March 2015, I worked full-time with Dartmouth professors Devin Balkcom and Thomas Cormen to develop material for Khan Academy's new algorithms curriculum:

https://www.khanacademy.org/computing/computer-science/algorithms

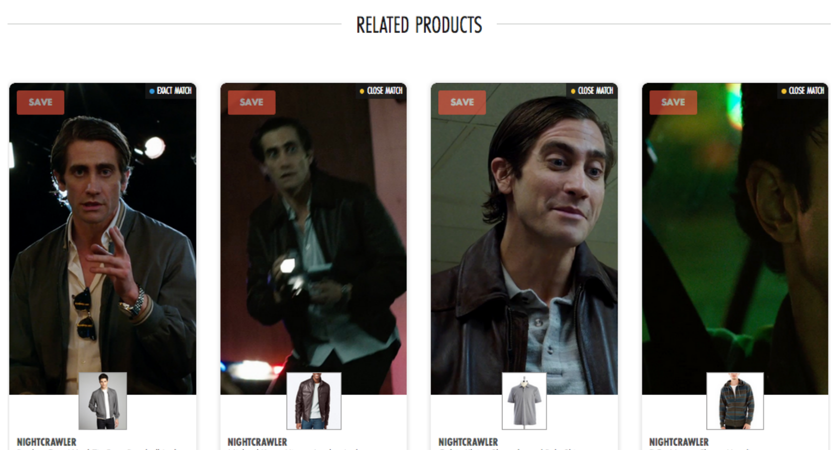
I explored topics to be included in future lessons of this introductory algorithms course, such as search algorithms, graph algorithms, stacks, queues, and trees. I wrote lessons, developed interactive demonstrations of algorithms, and designed assessments of student progress.

I wanted to make the demonstrations and challenges more interactive and intuitive than they previously had been. Here is an example of a challenge I created as part of a larger chapter not yet featured on the site. This challenge asks the student to implement depth-first search in an unknown maze and, upon completing the algorithm, shows the path taken and the dead-ends reached to solve the maze.

https://www.khanacademy.org/computer-programming/challenge-2-for-daniel/6661635214934016

2 Recommendation Engine

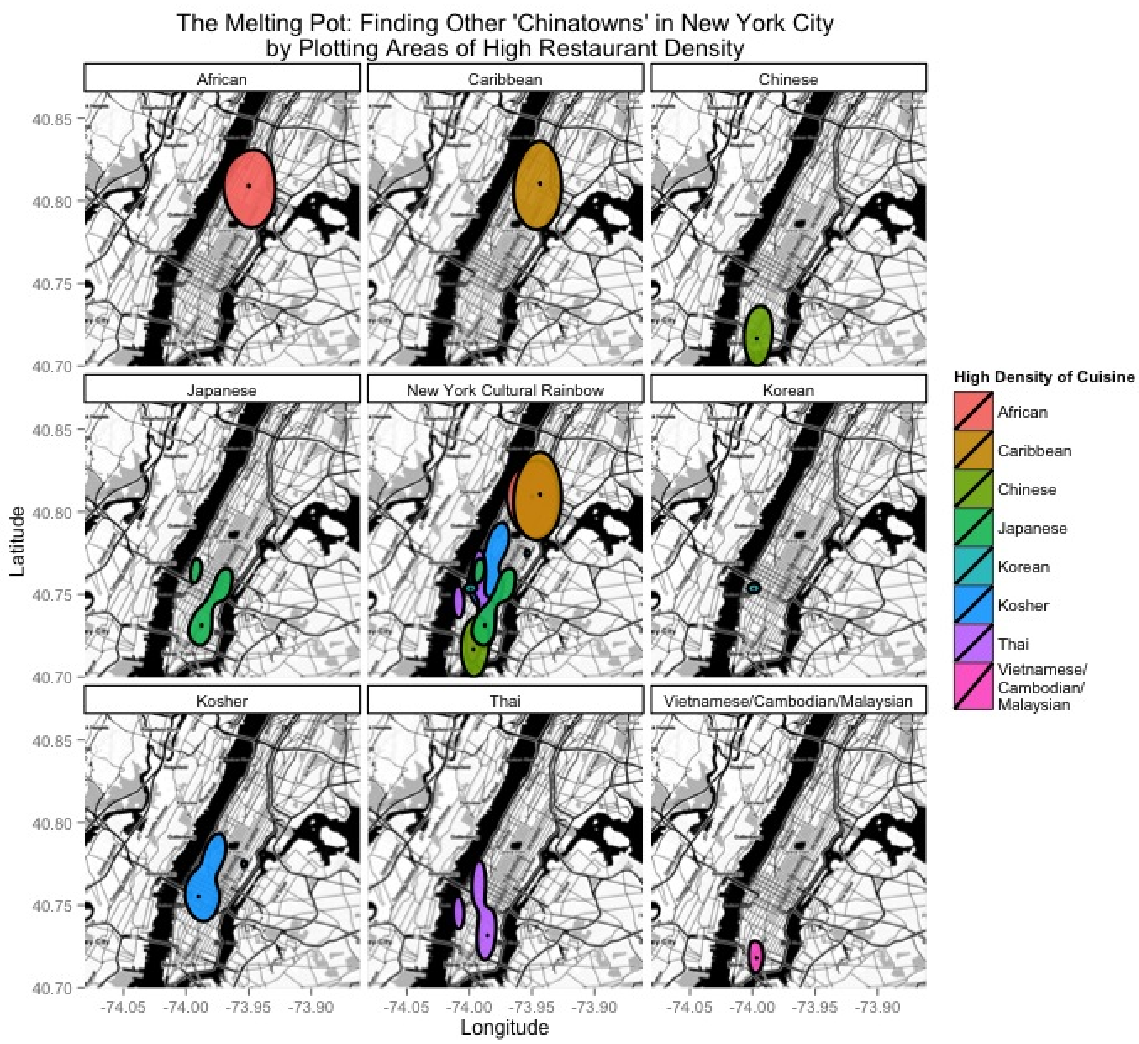
I worked at a small startup, TheTake, during the Summer of 2015. I designed and implemented a recommendation engine using item-based collaborative filtering ("People who viewed this also viewed...") and content-based similarity ("This actor also wore...") to provide recommendations for their shoppable movie platform.



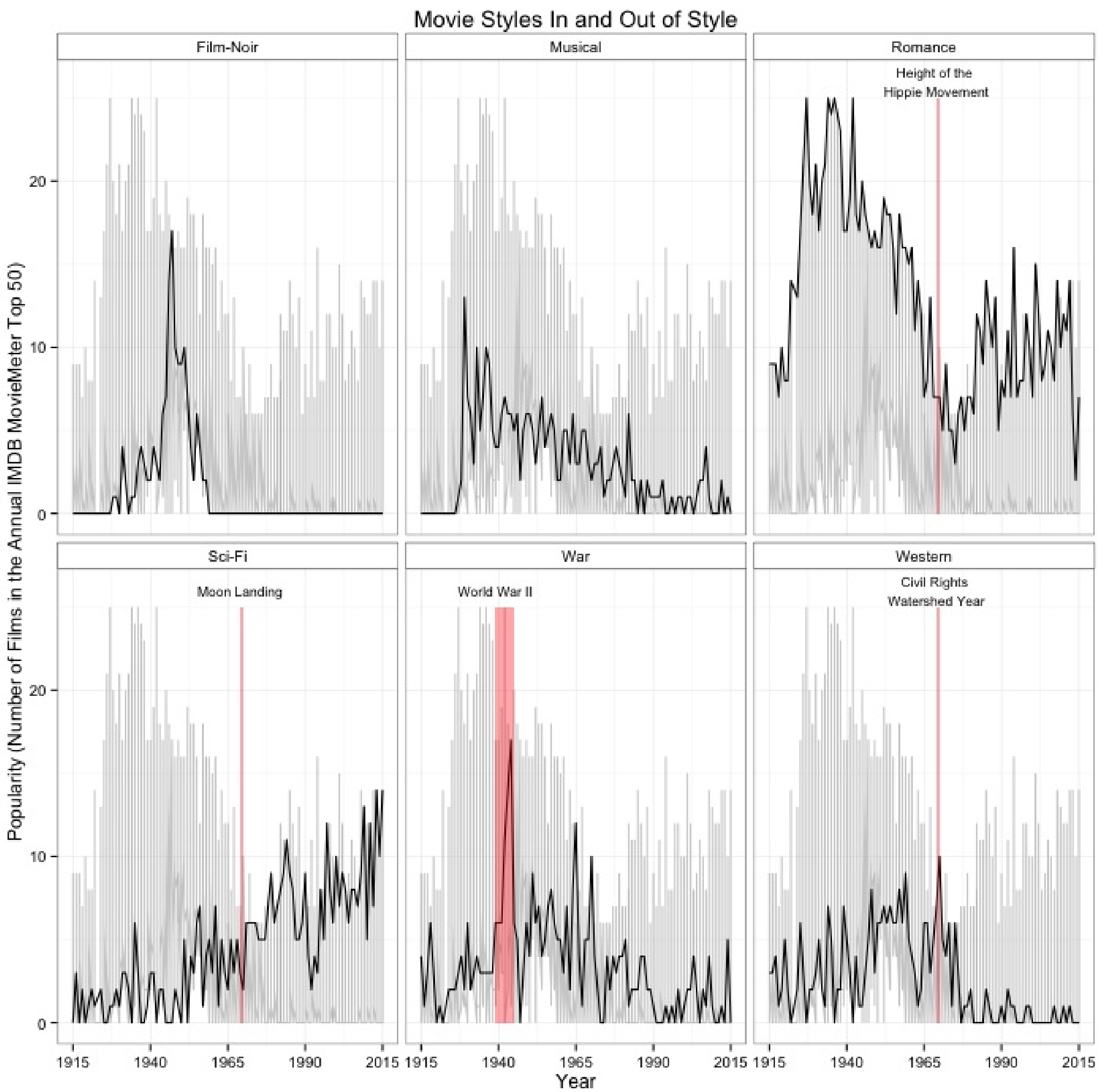
Though they have likely changed the algorithm over the past two years, you can see the recommendations under the "Related Products" tab on each individual product's page: https://thetake.com/

3 Data Visualization

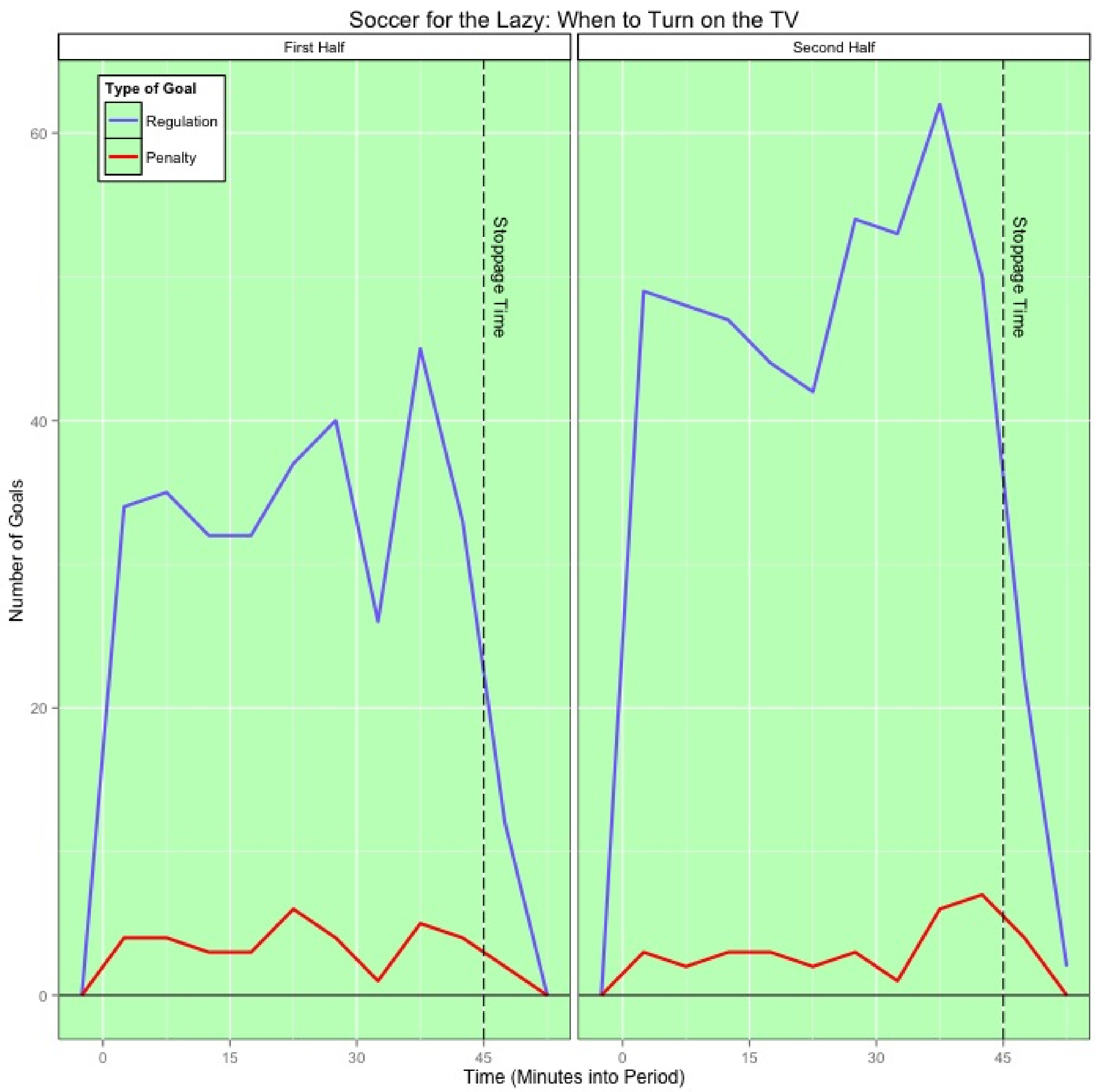
Using health inspection data tagged with location and cuisine type (defined as generally as it is in health inspection data) for all restaurants in New York City, I created a heatmap of various cuisines throughout Manhattan in order to try to identify pockets of culture, like Chinatown, within New York.

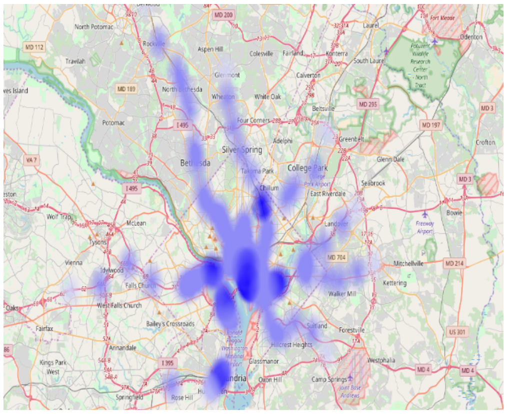


In order to examine American movie trends, I scraped data from each IMDB MovieMeter Top 50 list to track the popularity of different genres throughout the history of American cinema. These charts show the golden age of Noir, the slow death of the Musical, and some possible explanations for drastic changes (highlighted in red).

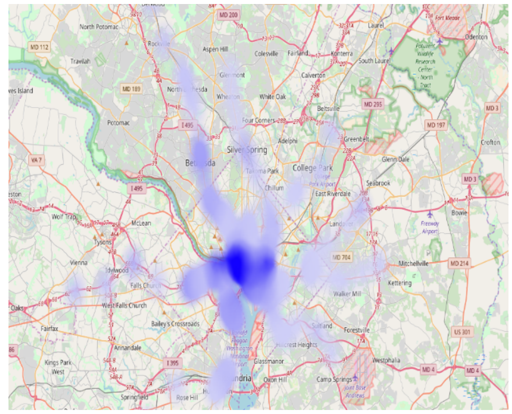


As a very fickle soccer fan who follows only the World Cup, I wanted to know the best time to tune in to catch the most exciting moments, rather than spread the energy of a few goals over a full 90-minute game. I scraped Wikipedia for every goal scored during every World Cup to create this visualization of how frequently goals are scored throughout the game.

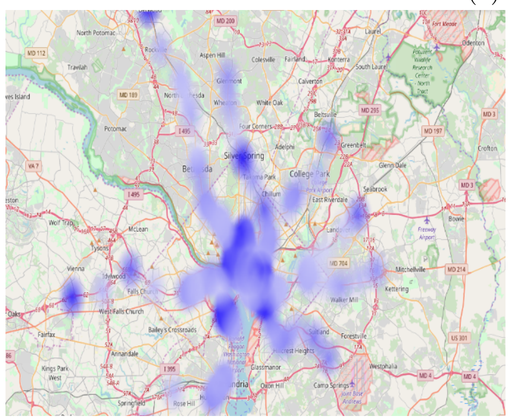




(a) Pure PageRank



(b) Morning Traffic



(c) Evening Traffic

Even before I worked at Google, I was fascinated by the PageRank algorithm, the system that defines how high up a page will appear in search results. I read the famous white paper and applied the same algorithm to the Metro subway system of my hometown, Washington D.C., partially to explore the algorithm and partially to explore traffic patterns in D.C.

These visualizations show how popular each subway stop would be if each station were a webpage linked to its neighbor stations (another webpage) by tracks in the same way that PageRank assigns popularity to webpages on the Internet. The pure PageRank visualization shows that the most visited pages are the ones linked to most often, or, in our case, the stations that serve as hubs between several sets of tracks. Just behind those in popularity are the stations adjacent to those, as they have inbound links from popular stations. The least popular stations are the ones at the ends of the lines, as they are each linked to by only a single, already unpopular, station.

I then modified the PageRank algorithm to take into account actual traffic patterns. Instead of connecting only adjacent stations by physical tracks, I used data from the Washington Area Metropolitan Transit Authority to connect each station to each other station with a weighted directed edge based on how many people commute from one station to any other given station on an average weekday. The morning visualization shows commuters traveling to work downtown, and the evening visualization shows them returning to notable suburban areas around Washington.

4 Tour Planning

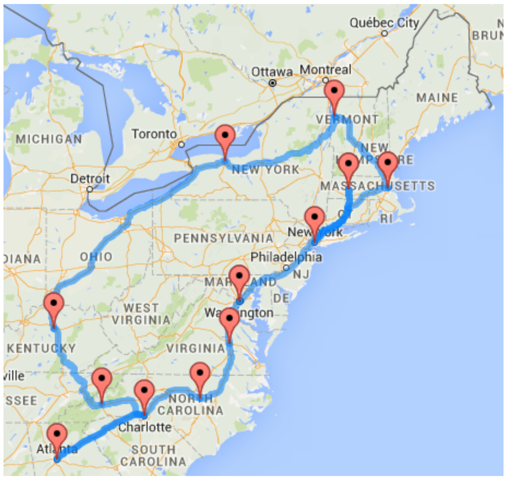
A few friends of expressed how difficult it was to plan their band's tour, which ended up being a perfect opportunity for me to explore genetic algorithms.

I first scraped data on the availability of preferred venues from the concert-tracking website Songkick, and then wrote a script using the Google Maps API to compute pairwise distances between the venues.

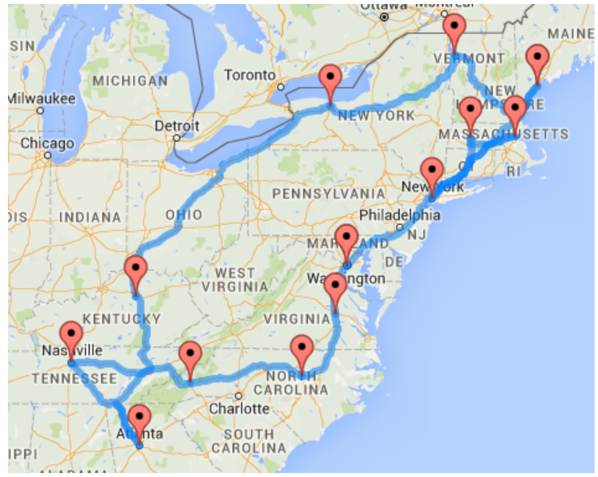
Venue	City	Capacity	15-Mar	16-Mar	17-Mar	18-Mar	19-Mar	20-Mar	21-Mar	22-Mar	23-Mar	24-Mar	25-Mar	26-Mar	27-Mar
Mercury Lounge	New York, NY	250	0	0	0	0	0	0	1	1	1	0	0	0	1
Red Room Café 939 Berklee College of Music	Boston, MA	200	1	1	1	1	0	1	0	0	0	0	0	1	1
Jammin Java	Vienna, VA	200	1	0	0	0	0	1	1	1	0	0	0	0	1
Club Café	Pittsburgh, PA	140	0	0	0	0	0	0	0	0	0	0	0	0	1
S.P.A.C.E.	Evanston, IL	250	1	0	0	0	1	0	1	0	0	0	0	0	1
Drunken Unicorn	Atlanta, GA	250	0	0	1	0	1	1	1	1	0	1	0	0	1
King's Barcade	Raleigh, NC	250	1	0	0	0	0	1	0	1	0	1	1	0	1
Rock & Roll Hotel	Washington, DC	400	0	0	0	0	0	0	1	1	1	1	1	0	1
Signal Kitchen	Burlington, VT	230	1	1	1	1	1	1	1	1	1	0	1	0	1
Boot and Saddle	Philadelphia, PA	150	0	0	0	0	0	1	1	1	0	0	0	0	0
Cosmic Charlie's	Lexington, KY	290	1	1	1	1	1	0	0	1	1	1	0	1	0
Gasa Gasa	New Orleans, LA	200	0	1	0	0	1	1	0	1	0	1	0	0	1
Exit/In	Nashville, TN	500	0	1	0	0	0	1	0	0	1	0	0	0	1
The Hi-Fi	Indianapolis, IN	175	0	1	0	0	0	0	0	1	0	1	0	0	0
Iron Horse Music Hall	Northampton, MA	250	0	0	0	0	0	1	1	0	1	0	1	0	0
Portland House of Music	Portland, ME	300	1	0	0	0	0	1	1	1	1	0	0	0	1
The Mothlight	Asheville, NC	250	0	1	1	0	1	1	1	1	0	0	0	0	0
The Broadberry	Richmond, VA	500	1	0	0	0	1	1	1	1	1	1	0	0	0
Water Street Music Hall	Rochester, NY	400	0	1	1	1	1	1	1	1	0	1	1	1	1
Independent Public Ale House	Charlotte, NC	140	1	1	1	1	1	1	1	1	1	1	1	1	1

waypoint1	Asheville,	Atlanta	Boston,	Burlington	Charlotte,	Evanston	Indianapolis	Lexington	Nashville,	New Orleans	New York	Northampton,	Philadelphia,	Pittsburgh,	Portland,	Raleigh,	Richmond,	Rochester,	Vienna,	Washington,
Asheville, NC	0	208	911	961	130	675	474	285	294	676	689	856	610	477	1010	247	373	754	458	471
Atlanta, GA	208	0	1084	1152	245	734	534	382	250	470	871	1026	782	685	1179	405	531	962	639	639
Boston, MA	911	1084	0	216	845	1001	949	925	1105	1526	215	104	310	572	112	723	553	393	452	439
Burlington, VT	961	1152	216	0	915	921	869	885	1075	1576	300	190	383	584	257	794	626	313	525	511
Charlotte, NC	130	245	845	915	0	774	575	400	409	713	632	790	545	448	944	167	293	725	400	400
Evanston, IL	675	734	1001	921	774	0	201	394	489	945	808	926	777	479	1100	828	817	620	713	715
Indianapolis, IN	474	534	949	869	575	201	0	189	289	819	708	874	643	360	1048	630	627	569	572	576
Lexington, KY	285	382	925	885	400	394	189	0	214	746	704	871	640	372	1024	494	492	584	530	536
Nashville, TN	294	250	1105	1075	409	489	289	214	0	532	884	1052	804	562	1204	539	614	773	652	666
New Orleans, LA	676	470	1526	1576	713	945	819	746	532	0	1305	1470	1225	1094	1623	877	999	1304	1073	1086
New York, NY	689	871	215	300	632	808	708	704	884	1305	0	161	97	369	314	509	340	334	239	226
Northampton, MA	856	1026	104	190	790	926	874	871	1052	1470	161	0	255	516	195	666	499	317	397	383
Philadelphia, PA	610	782	310	383	545	777	643	640	804	1225	97	255	0	304	408	422	253	341	153	139
Pittsburgh, PA	477	685	572	584	448	479	360	372	562	1094	369	516	304	0	671	502	344	284	241	242
Portland, ME	1010	1179	112	257	944	1100	1048	1024	1204	1623	314	195	408	671	0	819	652	492	550	536
Raleigh, NC	247	405	723	794	167	828	630	494	539	877	509	666	422	502	819	0	171	653	278	278
Richmond, VA	373	531	553	626	293	817	627	492	614	999	340	499	253	344	652	171	0	483	109	109
Rochester, NY	754	962	393	313	725	620	569	584	773	1304	334	317	341	284	492	653	483	0	380	384
Vienna, VA	458	639	452	525	400	713	572	530	652	1073	239	397	153	241	550	278	109	380	0	17
Washington, DC	471	639	439	511	400	715	576	536	666	1086	226	383	139	242	536	278	109	384	17	0

As a baseline, I found a tour using a simple greedy algorithm by starting in Boston and traveling each night to the nearest city with an unvisited available venue. Then, using a variety of constraints (never drive more than 600 miles a day, start and end in Boston, don't play the same venue twice, etc.) and heuristics (expected number of tickets sold based on venue size, price of gas based on miles driven, etc.), I implemented simulated annealing and designed a genetic algorithm to find a better tour. The optimal tour found by these algorithms, while not necessarily guaranteed to be the best possible tour, was still 20% more efficient than the simple greedy approach. By running these algorithms thousands of times, I was also able to compare the efficacy of the two metaheuristics and their randomized aspects.



(a) Tour Generated by Greedy Algorithm



(b) Best Tour Generated by Metaheuristics