빅데이터 분석과 시각화 개론 최종보고서
-서울시 대중교통 이용경향 분석 및 시각화-

201511045 김정균
201511058 김태연
201511104 신경식

서론

우리는 이번 프로젝트의 주제로 '서울시 대중교통 이용경향 분석'을 하였다.
누구나 '지옥철'이라는 말과 '출근 전쟁'이라는 말은 들어 봤을 것이다. 서울시에는 매우 많은 사람들이 밀집하여 있고 그 유동인구도 매우 많다. 그래서 실제로 서울시에서 지하철이나 버스를 타면 매우 붐비는 것을 어렵지 않게 경험하였을 것이다. 우리는 이러한 경험을 바탕으로 서울시 대중교통의 이용경향을 분석하면 개인적으로도 붐비는 시간대나 구간을 피할 수 있고, 정책적 측면에서도 대중교통 이용 인원이 너무 붐비는 것을 해결 할 수 있을 것이라고 판단하여서 이러한 주제를 정하게 되었다.
이번 프로젝트에서는 버스와 지하철의 혼잡도를 구해주어, 버스 한 대 또는 지하철 한 량의 탑승인원을 구해서 붐비는 구간을 확인하고, 버스와 지하철이 지나는 정류장 또는 역의 주소정보를 이용하여 구간 인구이동을 파악하는 것을 목표로 하였으나, 지하철의 경우는 혼잡도를 위한 데이터가 1~4 호선까지만 존재하여, 이를 이용해서는 유의미한 결과를 얻기 힘들다고 판단해서 지하철의 혼잡도를 구해주지 않고 버스의 혼잡도만을 이용하여 결과를 도출하였다.

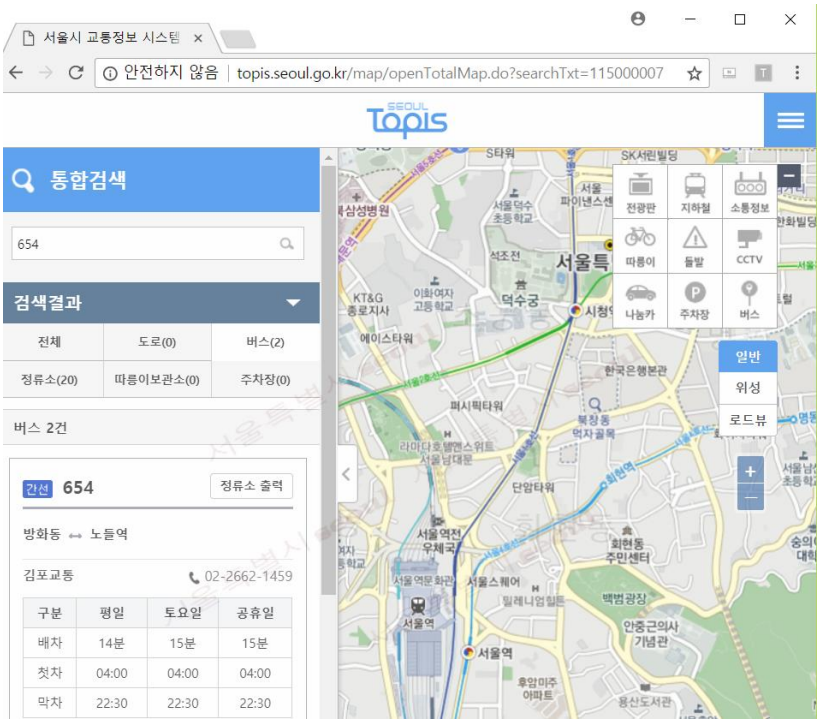
사용한 data set/ API

1. 서울시 버스 노선별/정류소별/시간대별 승·하차 인원 정보 file

	"사용년월"	노선번호	노선명	표준버스정류장ID	버스정류장ARS번호	역명	00시승차총승객수	00시하차총승객수	1시승차총승객수	1시하차총승객수	...	19시하차총승객수	20시승차총승객수	20시하차총승객수	21시승차총승객수	21시하차총승객수	22시승차총승객수	22시하차총승객수	23시승차총승객수	23시하차총승객수	등록일자
0	201710	6411	6411번(구로동~개포동)	121000107	22183	논현사거리.논현역5번출구	216	86	0	0	...	260	272	229	269	228	312	264	242	190	20171103
1	201710	6411	6411번(구로동~개포동)	121000017	22017	반포역	78	76	0	0	...	275	254	215	226	175	230	249	159	166	20171103
2	201710	6411	6411번(구로동~개포동)	121000019	22019	고속터미널	616	136	0	0	...	571	1148	369	1013	314	933	358	660	240	20171103
3	201710	6411	6411번(구로동~개포동)	121000021	22021	신반포역	70	82	0	0	...	257	221	164	190	139	182	205	89	105	20171103
4	201710	6411	6411번(구로동~개포동)	121000023	22023	구반포역	137	75	0	0	...	232	297	183	257	136	261	223	151	116	20171103

이 데이터는 서울 열린 데이터 광장에서 수집하였고, row 의 개수는 약 126 만개 정도이며, 서울시 버스의 노선별/정류소별/시간대별 승·하차 인원 정보이다.

2. 서울시 교통정보 시스템 사이트 crawling



서울시 교통정보 시스템 사이트 crawling 을 통해 노선별로 첫차/막차시간, 배차간격을 추출했다.

3. Google maps Geocoding API

```
https://maps.googleapis.com/maps/api/geocode/json?latlng=40.714224,-73.961452&key=YOUR_API_KEY
```


```
},
{
   "address_components" : [
      {
         "long_name" : "100-101",
         "short_name" : "100-101",
         "types" : [ "postal_code" ]
      },
      {
         "long_name" : "태평로1가",
         "short_name" : "태평로1가",
         "types" : [ "political", "sublocality", "sublocality_level_2" ]
      },
      {
         "long_name" : "중구",
         "short_name" : "중구",
         "types" : [ "political", "sublocality", "sublocality_level_1" ]
      },
      {
         "long_name" : "서울특별시",
         "short_name" : "서울특별시",
         "types" : [ "locality", "political" ]
      },
      {
         "long_name" : "서울특별시",
         "short_name" : "서울특별시",
         "types" : [ "administrative_area_level_1", "political" ]
      },
      {
         "long_name" : "대한민국",
         "short_name" : "KR",
         "types" : [ "country", "political" ]
      }
   ],
   "formatted_address" : "대한민국 서울특별시 중구 태평로1가",
   "geometry" : {
      "bounds" : {
         "northeast" : {
            "lat" : 37.5693753,
            "lng" : 126.9793525
         },
         "southwest" : {
            "lat" : 37.5638146,
            "lng" : 126.9741855
         }
      },
      "location" : {
         "lat" : 37.5675672,
         "lng" : 126.9772777
      },
      "location_type" : "APPROXIMATE",
      "viewport" : {
         "northeast" : {
            "lat" : 37.5693753,
            "lng" : 126.9793525
```


Google maps Geocoding API 는 주소를 GPS 로 바꿔주는 것과 GPS 를 주소로 바꿔주는 API 인데, 이중에서 GPS 를 주소로 바꿔주는 것을 사용하였다.

4. 서울시 지하철 호선별/역별/시간별 승·하차 인원 정보 file

	"사용월"	호선명	지하철역	04시-05시 승차인원	04시-05시 하차인원	05시-06시 승차인원	05시-06시 하차인원	06시-07시 승차인원	06시-07시 하차인원	07시-08시 승차인원	...	23시-24시 하차인원	00시-01시 승차인원	00시-01시 하차인원	01시-02시 승차인원	01시-02시 하차인원	02시-03시 승차인원	02시-03시 하차인원	03시-04시 승차인원	03시-04시 하차인원	작업일자
0	201710	경원선	가능	331	7	7630	1609	12558	4499	25256	...	7788	128	2368	0	0	0	0	0	0	20171103
1	201710	경원선	녹양	482	1	4753	900	9253	3182	17114	...	5802	72	2182	0	0	0	0	0	0	20171103
2	201710	경원선	양주	1014	3	6449	1598	14017	5672	34904	...	9935	46	3899	0	0	0	0	0	0	20171103
3	201710	경원선	덕계	8	0	2134	525	5420	1644	8654	...	3321	15	784	0	0	0	0	0	0	20171103
4	201710	경원선	덕정	1	0	4010	1129	8728	4130	23793	...	11029	128	2253	0	0	0	0	0	0	20171103

서울 열린 데이터 광장에서 수집하였고, 지하철 호선별/역별/시간별 승·하차 인원 정보이다.

5. 서울교통공사 지하철 역별 주소 file

	"연번"	호선	역명	도로명주소	지번주소	우편번호	전화번호
0	1	5	방화역	서울특별시 강서구 금낭화로 지하132 (방화동)	서울특별시 강서구 방화동 829-15	157-857	02-6311-5101
1	2	5	개화산역	서울특별시 강서구 양천로 22 (방화동)	서울특별시 강서구 방화동 846	157-220	02-6311-5111
2	3	5	김포공항역	서울특별시 강서구 하늘길 지하77 (방화동)	서울특별시 강서구 방화동 886	157-220	02-6311-5121
3	4	5	송정역	서울특별시 강서구 공항대로 지하33 (공항동)	서울특별시 강서구 공항동 29-5	157-811	02-6311-5131
4	5	5	마곡역	서울특별시 강서구 공항대로 지하163 (가양동)	서울특별시 강서구 가양동 530-6	157-805	02-6311-5141
5	6	5	발산역	서울특별시 강서구 공항대로 지하267 (가양동)	서울특별시 강서구 가양동 967-3	157-805	02-6311-5151
6	7	5	우장산역	서울특별시 강서구 강서로 지하262 (화곡동)	서울특별시 강서구 화곡동 1089-3	157-010	02-6311-5161
7	8	5	화곡역	서울특별시 강서구 화곡로 지하168 (화곡동)	서울특별시 강서구 화곡동 1089-54	157-010	02-6311-5171
8	9	5	까치산역	서울특별시 강서구 강서로 지하54 (화곡동)	서울특별시 강서구 화곡동 662-5	157-010	02-6311-5181
9	10	5	신정역(은행정)	서울특별시 양천구 오목로 지하179 (신정동)	서울특별시 양천구 신정동 1090	158-070	02-6311-5191

서울교통공사에서 수집하였고, 지하철 역별 주소 정보이다.

6. 자치구별 세대수

	기간	구분	단지수	동수	세대수	전용면적별 세대수	전용면적별 세대수.1	전용면적별 세대수.2	전용면적별 세대수.3	전용면적별 세대수.4	전용면적별 세대수.5
0	기간	구분	단지수	동수	세대수	40㎡ 이하	40~60㎡ 이하	60~85㎡ 이하	85~102㎡ 이하	102~135㎡ 이하	135㎡ 초과
1	2015	합계	4,214	19980	1524297	104,025	269,152	506,819	232,129	292,066	120,106
2	2015	종로구	58	235	11033	8875926	2,500	4,171	2,553	NaN	NaN
3	2015	중구	58	203	20431	5046	1,485	1,848	7,512	2,876	1,664
4	2015	용산구	111	454	30931	813	2,019	8,605	7,156	6,743	5,595
5	2015	성동구	121	641	48524	-	14,234	23,536	1,128	8,295	1,331
6	2015	광진구	109	351	27481	238	6,072	15,808	458	2,176	2,729
7	2015	동대문구	151	658	52188	1,231	2,291	12,487	10,972	17968	7239
8	2015	중랑구	147	593	46881	2902	7535	14513	14852	5830	1249
9	2015	성북구	161	992	68931	561	6849	28414	8399	19445	5263

서울 열린 데이터 광장에서 수집하였고, 서울시 자치구별 세대수 정보이다.

7. 자치구별 통학·통근 인원

	기간	자치구별	12세이상 인구	통근통학 안함	계	현재살고 있는 동	같은 자치구 내 다른 동	같은 시내 다른 자치구	다른 시·도
0	2015	합계	8613785	2864276	5749509	2174764	404422	2437802	732521
1	2015	종로구	136954	45668	91286	42875	7854	33821	6736
2	2015	중구	109428	37965	71463	34444	6307	25847	4865
3	2015	용산구	192970	66719	126251	49195	8204	55996	12856
4	2015	성동구	257235	83441	173794	63614	10311	83364	16505
5	2015	광진구	320108	97368	222740	82624	12768	104367	22981
6	2015	동대문구	318474	109094	209380	87380	14662	89426	17912
7	2015	중랑구	358990	121791	237199	88317	13632	107009	28241
8	2015	성북구	400299	134751	265548	97574	18251	127197	22526
9	2015	강북구	286164	106223	179941	62896	9894	90978	16173

서울 열린 데이터 광장에서 수집하였고, 서울시 자치구별로 같은 동 내, 같은 자치구 내 다른 동, 같은 시내 다른 자치구, 다른 시·도 간 통학·통근 인원 정보이다.

Data Set 을 이용하여 구해준 data

1. 버스노선별 어떤 순서로 정류소를 들리는지 정리한 data
   서울시 대중교통 정보사이트 Crawling 을 통해서 노선별로 어떤 순서로 정류소를 들리는지 정리했다.

2. 버스정류장의 주소 구하기
   버스의 혼잡도와 상관관계 분석을 서울 자치구별로 나누어서 분석을 하였는데, 그러기 위해서 각 노선의 정류소가 어떤 자치구에 위치해 있는지 알아야 하기 때문에 이를 구하는 과정이 필요하다. 서울시 대중교통 정보사이트(m.bus.go.kr/)에서 Crawling 을 통해서 버스 노선별로 GPS 정보를 추출하였고, 이 GPS 정보와 Google maps 역지오코딩 API 를 이용하여, GPS 정보를 주소 정보로 변환시켜주었다. (한 개의 API 당 하루에 2500 개의 GPS 정보를 주소로 바꿀 수 있고, 약 6 개의 API key 를 사용하였다.) API 에 의해 구한 주소 data 가 python list 형태로 복잡하게 나타나기 때문에 이를 우리가 필요한 자치구 이름만 뽑아내서 버스 정류소를 자치구별로 정리하였다. 다음은 정리해준 파일의 일부이다.

9101	115000234	Gangseo-gu
9102	115000244	Gangseo-gu
9103	115000920	Gangseo-gu
9104	115000921	Gangseo-gu
9105	115000038	Gangseo-gu
9106	121000194	Gangnam-gu
9107	121000006	Gangnam-gu
9108	122000093	Gangnam-gu
9109	122000094	Gangnam-gu
9110	122000087	Gangnam-gu
9111	122000107	Gangnam-gu
9112	122000109	Gangnam-gu
9113	122000113	Gangnam-gu

3. 버스의 혼잡도 구하기

버스노선별로 정류소간 거리를 구해주기 위해서 전에 구해 놓았던 버스정류장별 GPS 정보와 버스 노선별 버스정류장 경로 정보를 이용하였다. 이를 이용해서 실제 도로를 통해 가는 거리는 구하기 어려울 것이라고 판단하여 각각 앞뒤의 정류장들은 직선으로 연결 되어있다고 가정을 하였다. 지구 반지름은 약 6371.0km 라고 가정을 하여 위도 경도 좌표사이의 거리를 계산을 하여 정류장 사이의 거리를 구해주었다. 정류장 사이의 거리를 구해준 후에는 버스의 혼잡도를 구해주기 위해서 버스 정류소간 소요시간을 구해주었다. 서울시 통계에 의하면 서울시 버스의 평균 속도는 약 20km/h 이지만, 가정에서 모든 거리를 직선거리로 가정을 해주었고, 정류장별 버스 대기시간 등은 고려를 해주지 않았기 때문에 버스의 평균 운행 속도는 15km/h 라고 가정을 하여 정류장별 소요시간을 구해주었다. 다음 그림은 버스의 정류장별 소요시간을 구한 결과의 일부이다.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	100100048														
2	0	1.287023	2.190121	3.111006	4.095133	5.203884	6.302433	7.290253	8.67709	9.754214	12.56292	15.71471	18.34857	19.71035	21.69
3	100100017														
4	0	0.817789	1.609731	3.877506	5.025537	6.110407	7.676491	8.784072	9.948056	11.91811	12.78922	13.8577	16.44858	19.24453	21.65
5	124000010														
6	0	1.020518	2.439388	3.588767	4.619149	5.823388	8.811983	10.87362	13.03883	14.03151	14.97333	16.78538	18.94594		
7	124900003														
8	0	1.443788	2.325919	4.078536	4.547176	5.451811	6.314755	7.201296	8.02071	9.744768	10.69889	11.97572	13.24984	14.2428	15.32
9	100100226														
10	0	0.451562	1.695324	3.035335	4.566313	5.738214	6.615025	7.616662	8.820747	9.813428	10.75525	12.5673	14.72785	16.04947	17.5
11	100100295														
12	0	0.822482	2.457027	3.064176	4.407291	4.820438	5.589422	6.303271	6.89369	7.434225	8.813597	10.16976	10.68615	11.26994	11.76
13	116900016														
14	0	0.540955	0.98555	1.337623	1.735483	2.633502	3.873541	4.266546	5.451038	6.294838	7.34289	8.233309	9.386179	9.991325	11.21
15	100100392														
16	0	2.603651	4.01647	4.609992	6.336744	7.222682	8.321325	9.1857	10.21391	11.5801	14.28802	15.61087	16.62933	19.08533	21.18
17	121900005														

위의 그림에서 각각 먼저 버스 노선 ID 가 써져 있고, 그 바로 아래의 row 에 버스 정류장 순서대로 첫번째 정류장으로부터 걸리는 시간을 분단위로 적어주었다.

다음으로는 이 정류장별 소요시간을 바탕으로 하여 버스노선별 정류소별 시간별 들리는 버스 수를 구해주었다. 버스노선별로 첫차시간과 막차시간, 배차간격의 정보를 가지고 있기 때문에 이 정보와 버스노선별 들리는 정류장 순서대로 정류장별 소요시간을 이용하여 각각의 버스 정류장에서 시간당 특정 버스 노선이 몇 대가 지나가는지를 구해주었다. 다음 그림은 시간당 버스 노선 별 정류장에 시간당 버스 노선이 몇 대가 지나가는지를 구한 데이터의 일부이다.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	100100044														
2	106000319	0	0	0	0	6	6	6	6	6	6	6	6	6	
3	106000242	0	0	0	0	6	6	6	6	6	6	6	6	6	
4	106000442	0	0	0	0	6	6	6	6	6	6	6	6	6	
5	106000240	0	0	0	0	6	6	6	6	6	6	6	6	6	
6	106000238	0	0	0	0	6	6	6	6	6	6	6	6	6	
7	106000235	0	0	0	0	6	6	6	6	6	6	6	6	6	
8	106000233	0	0	0	0	6	6	6	6	6	6	6	6	6	
9	106000231	0	0	0	0	6	6	6	6	6	6	6	6	6	
10	106000049	0	0	0	0	6	6	6	6	6	6	6	6	6	
11	106000431	0	0	0	0	6	6	6	6	6	6	6	6	6	
12	106000433	0	0	0	0	5	6	6	6	6	6	6	6	6	
13	106000434	0	0	0	0	5	6	6	6	6	6	6	6	6	
14	106000001	0	0	0	0	5	6	6	6	6	6	6	6	6	
15	106000073	0	0	0	0	5	6	6	6	6	6	6	6	6	
16	106000075	0	0	0	0	5	6	6	6	6	6	6	6	6	
17	106000076	0	0	0	0	5	6	6	6	6	6	6	6	6	
18	106000079	0	0	0	0	5	6	6	6	6	6	6	6	6	
19	106000081	0	0	0	0	5	6	6	6	6	6	6	6	6	
20	106000083	0	0	0	0	4	6	6	6	6	6	6	6	6	
21	106000085	0	0	0	0	4	6	6	6	6	6	6	6	6	

데이터에 대해서 간략하게 설명을 하면, 먼저 각 노선별로 버스노선 ID 를 써준 후, 그 밑에부터 그 버스가 들리는 버스 정류장 순서대로 row 가 쓰여진다. 각 row 에서 첫번째 성분은 버스정류장 ID 를 나타내며, 다음부터는 0 시부터 시작하여 한시간 단위로 각 시간대에 지나가는 버스의 수를 나타낸다.

다음으로는 지금까지 구해준 데이터를 이용하여 버스노선별 정류소별 시간별 버스 한 대당 승·하차 인원을 구해주었다. raw data 로 이용한 버스 승·하차 정보가 월별로 버스 정류장별 시간대별 승·하차인원에 대한 정보를 주고 있기 때문에 앞서 구한 데이터에서 정류장, 버스 노선별로 각 시간대에 몇 대의 버스가 오는지 알 수 있기 때문에 버스 한 대당 승·하차 인원을 구해줄 수 있었다. 그리고 raw data 의 기준이 월별 데이터였으므로 30.3 일로 나누어 주어 일별 버스 한 대당 승·하차 인원을 근사하여 구해주었다.

마지막으로 노선별 시간별 버스혼잡도를 구해주었다. 버스 한 대 당 승·하차인원을 구해주었기 때문에, 버스가 노선을 따라가면서 승·하차인원 만큼을 누적으로 더해주고, 하차 인원만큼을 누적으로 빼 주면서 버스혼잡도를 구해주었다.
구해준 데이터는 csv 파일으로 버스노선별로 첫 줄에 버스 노선 ID 를 써주고, 다음 줄부터는 한 row 에 버스 정류장의 버스 대수 별 혼잡도를 써주었고, row 들은 버스가 들리는 정류장 순서대로 써주었다. 즉, 버스 ID 가 한 번 나오면 다음 버스 ID 가 나올 때까지 하나의 column 이 한 대의 버스가 정류장 순서대로 운행하는 것을 나타내게 된다. 최종적으로는 약 36600 줄의 데이터를 얻어주었다. 하지만, 이 과정에서 몇가지의 문제가 생겼는데 첫번째로는 중간에 혼잡도가 음의 값을 가지는 구간이 생기었다. 음의 혼잡도가 나온 이유에 대해서 생각을 해보면, 한달 단위의 통계 데이터를 이용해서 일별 데이터로 추측하여 구해준 것이기 때문에 일별로 승차인원과 하차인원이 정확하게 나오지 않는 구간이 있을 수 있을 것이라고 판단을 하였다. 음의 혼잡도가 나오는 구간에 대해서는 구해준 혼잡도에서 음의 값이 나오면 모두 0 으로 처리를 해주도록 처리를 하여 해결해주었다. 다음 그림은 음의 혼잡도를 처리해준 후의 데이터의 일부이다.

137	13	17	16	0	0	0	25	26	29	9	17	23	20
138	13	17	15	0	0	0	25	26	29	9	17	23	20
139	13	17	15	0	0	0	25	26	28	8	16	22	19
140	13	17	15	0	0	0	25	26	28	8	16	22	19
141	13	16	15	0	0	0	25	25	28	8	16	21	18
142	13	16	15	0	0	0	24	25	28	8	16	21	18
143	100100300												
144	1	1	1	1	1	1	1	1	1	1	1	1	1
145	1	1	1	1	1	1	1	1	1	3	3	3	3
146	1	1	1	1	1	2	2	2	2	6	6	6	6
147	1	1	1	1	1	2	2	2	2	8	8	8	8
148	1	1	1	1	2	3	3	3	3	11	11	11	11
149	1	1	1	1	2	3	3	3	3	12	12	12	12
150	1	1	1	1	2	3	3	3	4	13	13	13	13
151	1	1	1	1	2	3	3	3	4	13	13	13	13
152	1	1	1	1	2	3	3	3	4	13	13	13	13
153	1	1	1	1	2	3	3	3	3	12	12	12	12

그림에서 볼 수 있듯이, 두번째 문제로는 버스가 종점에 도착을 해도 버스의 혼잡도가 0 이 되지 않는 경우가 많았다. 이 이유에 대해서 생각을 해보면, 일단 raw data 의 추산 방법을 생각해볼 수 있을 것이다. 이 데이터의 추산 방법은 교통카드 리더 기록을 이용하는 것으로, 승차할 때 읽혀진 기록을 통해 승차 인원을 추산하고, 내릴 때 카드를 한 번 더 찍는 것으로 하차 인원을 추산한다. 하지만, 실제로는 현금이나 무임승차를 하는 경우도 많을 것이고, 이 경우는 하차인원은 집계가 사실상 불가능하다. 또한, 서울시에서는 버스를 내릴 때 항상 교통카드를 찍고 있지만, 실제로는 카드를 찍지 않고 내리는 사람도 꽤 많을 것으로 판단하였다. 실제로 데이터를 보면 어느 시점에서부터 혼잡도 인원이 계속 일정하게 유지되어 종점까지 도착하는 경우를 확인할 수 있었고 이러한 영향때문에 조금은 부정확한 결과가 나왔다고 판단하였다. 이를 수정해주기 위해서는 마지막까지 남아있는 인원은 적당히 고르게 버스 운행 구간내에서 한 명씩 추가로 하차 시켜주었다. 이 방법에 대해 조금 더 자세히 설명을 하면, 마지막 정류장에서 혼잡도가 0 이 아닐 때 그 숫자로 버스가 들리는 정류장의 수를 나누어 준 몫을 구해주어 그 몫의 정수배에 해당되는 정류장에서 추가적으로 한 명씩 더 하차를 시켜서 종점에서의 인원을 0 명으로 만들어주었다. 이렇게 하면서 누적으로 인원이 빼지는 경우가 생겨서 데이터의 전체적인 혼잡도의 크기가 조금 작아 지기는 했지만, 이 데이터가 더 정확한 근사를 나타내는 데이터라고 생각하여 이 데이터를 사용하기로 하였다. 종점의 혼잡도를 보정해준 결과를 보면 다음과 같다.

137	0	1	1	0	0	0	1	1	2	1	1	2	2
138	0	1	0	0	0	0	1	1	2	1	1	2	2
139	0	1	0	0	0	0	1	1	1	0	0	1	1
140	0	1	0	0	0	0	1	1	1	0	0	1	1
141	0	0	0	0	0	0	1	0	0	0	0	0	0
142	0	0	0	0	0	0	0	0	0	0	0	0	0
143	100100300												
144	1	1	1	1	1	1	1	1	1	1	1	1	1
145	1	1	1	1	1	1	1	1	1	3	3	3	3
146	1	1	1	1	1	2	2	2	2	6	6	6	6
147	1	1	1	1	1	2	2	2	2	8	8	8	8
148	1	1	1	1	2	3	3	3	2	11	11	11	11
149	1	1	1	1	2	2	2	2	2	12	12	12	12
150	1	1	1	1	2	2	2	2	3	13	13	13	13
151	1	1	1	1	2	2	2	2	3	12	12	12	13

다음으로는 이 데이터의 상관관계를 확인해주기 위해서 혼잡도를 자치구별, 시간별로 다시 정리해주었다. 일단은 버스 노선에 대해서 데이터는 버스의 운행 경로 순서이기 때문에 버스 정류장의 주소 정보를 통해서 해당 정류장이 어느 구에 있는 정류장인지를 구해주었고, 이를 시간대별로 나누어 주기 위해서는 기존에 버스노선, 정류소별 시간별 들리는 버스 수의 데이터를 이용하여 같은 row 에서 그 수만큼 씩을 더해주어서 구해주었다. 이를 pandas 를 이용하여 상관관계를 분석할 때 가공하기 좋은 형태로 자치구별 시간대별로 분리하여 총 25 개의 row 를 가진 csv 파일로 만들어주었다. 그 결과를 보면 다음과 같다.

	0 ~ 1시	1 ~ 2시	2 ~ 3시	3 ~ 4시	4 ~ 5시	5 ~ 6시	6 ~ 7시	7 ~ 8시	8 ~ 9시	9 ~ 10시	...	14 ~ 15시	15 ~ 16시	16 ~ 17시	17 ~ 18시	18 ~ 19시	19 ~ 20시	20 ~ 21시
구 이름																		
Dobong-gu	9105	3825	524	265	6619	13432	27950	58403	67755	53647	...	46796	49971	55594	59600	63595	55269	46355
Dongdaemun-gu	9578	1724	866	551	12192	31228	50059	90039	116493	94175	...	82470	84240	93105	99914	111624	103797	82682
Dongjak-gu	9221	2625	1390	590	4964	21873	43903	80391	97555	77833	...	68406	70196	79129	85254	100557	94119	74486
Eunpyeong-gu	9579	3498	824	224	8371	15389	34620	66737	78800	64462	...	53302	58945	66417	70482	74470	64445	53110
Gangbuk-gu	10952	3935	519	738	16531	33480	50614	83329	92893	74499	...	68605	73214	81239	87057	90344	84161	72725
Gangdong-gu	3871	1387	779	435	4406	10656	24363	49823	53953	41429	...	37847	40476	47122	50873	54687	47054	39822
Gangnam-gu	17911	2876	1321	1398	7914	28761	55210	109256	158404	127679	...	106682	110860	123868	131843	152078	145372	11380
Gangseo-gu	6609	1351	391	221	10370	19120	33412	72074	88726	64599	...	61404	66079	74475	77205	87162	76243	61133
Geumcheon-gu	3670	282	105	411	8224	22425	34759	71592	86261	58027	...	45925	48475	55792	63034	75302	67937	51561
Guro-gu	5357	299	265	372	10813	18610	31022	64876	76229	56937	...	52840	56168	63360	67371	74337	67314	54767
Gwanak-gu	4059	895	503	1022	11392	30479	55165	105215	137778	112599	...	83563	85570	95941	104503	117697	114374	94965
Gwangjin-gu	6017	1609	1626	769	4778	13455	26610	44232	55424	45224	...	38616	40859	46079	49514	57764	56127	46803
Jongno-gu	18470	4370	1801	1687	8363	31611	47257	87584	110955	96528	...	95300	98675	107898	113902	127412	112554	92665
Jung-gu	8346	3028	1391	1101	3303	18217	23216	39370	54526	47865	...	51394	51417	54741	57956	65810	66448	50704
Jungnang-gu	5689	2837	807	721	9492	17427	32474	66315	75905	56888	...	50845	54023	61091	64208	69531	63449	51286
Mapo-gu	6903	1707	992	667	4797	17342	32109	66136	90714	71563	...	66077	68407	73568	78944	91876	83060	65027
Nowon-gu	4902	2140	769	326	7953	21091	37413	76701	83695	68366	...	65819	71245	81849	84510	84935	74316	62263
Seocho-gu	19152	6058	2672	1437	4790	20902	44012	89859	127117	98814	...	85347	86710	95489	99035	118279	105078	85684
Seodaemun-gu	12626	2994	934	624	9902	27265	52406	96882	114738	94980	...	84069	88100	99129	105583	114664	98088	82610
Seongbuk-gu	15124	4520	713	726	16229	36637	59396	108018	121279	98633	...	89956	94581	106889	114206	120140	107936	93674
Seongdong-gu	4160	759	841	509	4087	17128	25098	40730	53814	42651	...	37240	39110	42440	44863	49375	48846	39178
Songpa-gu	11629	1159	1330	397	10968	26744	46477	94811	128982	97664	...	76375	80757	92251	99936	113565	104548	84537
Yangcheon-gu	5725	625	432	211	8552	18098	35552	72227	80073	61570	...	55702	58923	68130	72453	80649	69979	56708
Yeongdeungpo-gu	7866	941	1159	540	10594	29219	40410	79435	103800	76267	...	74086	77864	84937	93731	114355	105495	78779
Yongsan-gu	8551	3300	773	368	2821	15820	29513	61578	82015	66262	...	63423	64117	70463	75176	83863	79353	60089

4. 시간별 자치구 간 버스 이동인원.
   버스 노선별 어떤 순서로 정류소를 들리는지 정리한 data 와 자치구별 버스 정류소 data 를 이용해서 자치구가 바뀌는 두 정류소의 쌍(버스 및 혼잡도 정보 포함)을 추출하고, 자치구가 바뀔 때 어떤 시간대에 해당하는지를 첫차 시간, 배차간격, 버스 정류소 간 소요시간을 이용하여 계산했다. 아래 사진은 9 시~10 시에 자치구별 이동인원을 나타낸 데이터이다. 각 row 는 출발한 구, column 은 도착한 자치구를 의미한다.

	Geumchec	Guro-gu	Gwanak-gu	Seocho-gu	Dongjak-g	Nowon-gu	Mapo-gu	Seodaemu	Gangseo-g	Yangcheor	Yeongdeu	Seongbuk
Geumchec	0	764	1433	0	0	0	0	0	0	0	0	0
Guro-gu	742	0	0	0	0	0	0	0	0	461	808	0
Gwanak-gu	337	112	0	788	1515	0	0	0	0	0	504	0
Seocho-gu	0	0	0	0	1231	0	0	0	0	0	0	0
Dongjak-g	0	0	1848	952	0	0	0	0	0	0	2441	0
Nowon-gu	0	0	0	0	0	0	0	0	0	0	0	325
Mapo-gu	0	0	0	0	0	0	0	1672	160	0	433	0
Seodaemu	0	0	0	0	0	0	1657	0	0	0	0	0
Gangseo-g	0	0	0	0	0	0	373	0	0	3267	190	0
Yangcheor	0	747	0	0	0	0	0	0	2547	0	1252	0
Yeongdeu	0	783	0	0	2304	0	772	0	362	193	0	0
Seongbuk	0	0	0	0	0	184	0	0	0	0	0	0
Gangbuk-g	0	0	0	0	0	57	0	0	0	0	0	1577
Songpa-gu	0	0	0	0	0	0	0	0	0	0	0	0
Eunpyeong	0	0	0	0	0	0	215	1478	0	0	0	0

5. 지하철의 총 승·하차인원

지하철의 경우는 앞서서 언급했듯이, 데이터의 부족으로 혼잡도는 구해주지 못했으며, 기존의 데이터를 자치구별로 분류를 해주어 상관관계를 확인해주기 위한 데이터를 만들어주었다. 일단 raw data 로 가지고 있는 지하철 호선별 역별 시간별 승·하차 인원을 이용해서 이 지하철 역을 자치구별로 분류해주었고, pandas 를 이용하여 비교하기 편한 형태로 만들어주었다. 역을 자치구별로 분류해주는 방법으로는 서울시에서 제공하고 있는 호선별 전화번호, 주소 데이터를 이용하여 1~8 호선까지만 분류해주었다. 그리고 이를 총 승·하차인원, 출근 퇴근시간의 승·하차인원으로 가공해주었으며, 그 결과를 보면 다음과 같다.

	구이름	총 승하차 인원 합	출근시간 승하차인원 합	퇴근시간 승하차 인원 합
0	강남구	1058259900	248397377	241745320
1	강동구	200847735	51047541	41132067
2	강북구	92104750	20817136	19938892
3	강서구	292271415	66837633	61940377
4	관악구	330370770	75668196	71794304
5	광진구	329328968	62659516	74245419
6	구로구	557631698	131920563	124897404
7	금천구	214819102	60437025	53642453
8	노원구	375809448	84859404	77278793
9	도봉구	165463461	38703457	33096336
10	동대문구	106277117	22720214	19959519
11	동작구	421261440	92920436	92102100
12	마포구	600808439	99974623	138972748
13	서대문구	100757033	25105187	20538462
14	서초구	516802260	100957919	114520098
15	성동구	283246912	71199534	61440541
16	성북구	180608274	41607461	36602633
17	송파구	621164048	136556423	134385749
18	양천구	89611009	21220221	19646146
19	영등포구	415135996	103068394	95990159
20	용산구	326998264	64203845	72714457
21	은평구	347663470	85620755	73098627
22	종로구	652341022	119146119	136498192
23	중구	728108446	156018683	166335399
24	중랑구	147384590	35959547	31258788

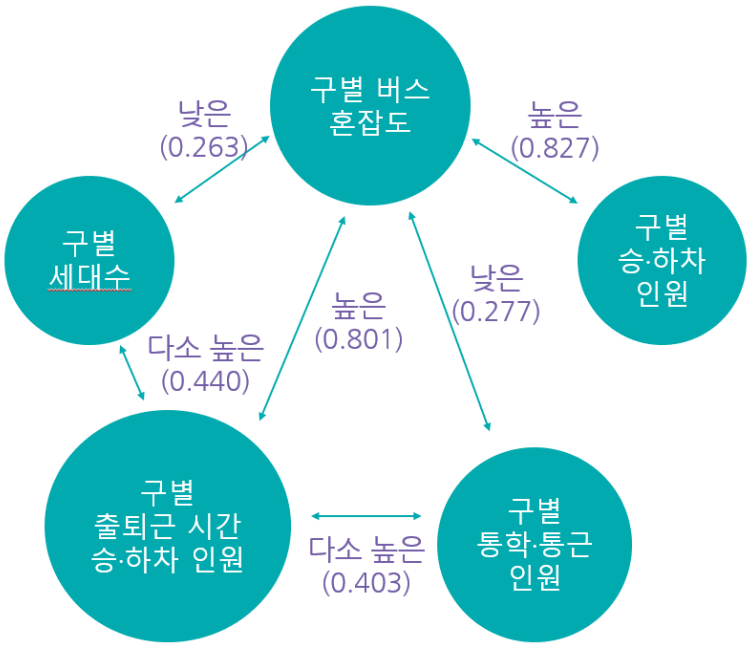
연관 데이터와의 상관관계 분석

우리가 구해 준 data 가 얼마나 유의미한 데이터인지 확인하기 위해 관련 데이터인 자치구별 세대수, 통학·통근 인원과의 상관관계를 구해보았다. 먼저 버스의 경우, 자치구별 승·하차 인원을 바탕으로 구해준 자치구별 버스 혼잡도의 경우 상관계수가 0.827 정도로 높은 상관관계를 나타내는데, 이는 데이터 처리 과정이나 가정에서 데이터 왜곡이 크게 일어나지 않았음을 나타낸다고 할 수 있다.

다음으로 혼잡도와 상관이 있을 것으로 예상되는 데이터인 자치구별 세대수와 자치구별 통학·통근 인원과의 상관관계를 구해보았는데, 각 0.263, 0.277 로 낮은 상관관계를 보임을 알 수 있었다. 그래서 전체 시간대의 혼잡도의 합과 상관관계를 찾기보다는 출퇴근 시간에 한해서만 구해준다면 좀 더 연관성이 있을 것이라고 생각하여 출퇴근 시간의 승·하차 인원과의 상관관계를 구해본 결과, 각 0.440, 0.403 으로 다소 높은 상관관계가 나타남을 알 수 있었다.

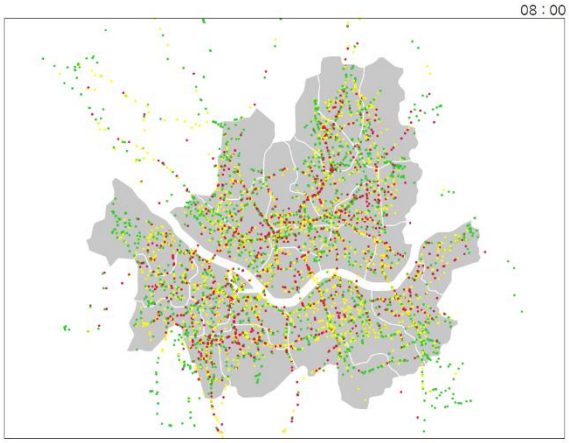
다음으로 지하철의 경우는 승·하차 인원과 자치구별 세대수, 자치구별 통학·통근 인원과의 상관관계가 거의 없음을 보였는데, 그 이유로 지하철역이 거주지보다는 도심 지역 위주로 위치해 있어서 세대수나 통학·통근 인원과는 상관관계가 낮게 나왔을 것이라고 추측하고 있다.

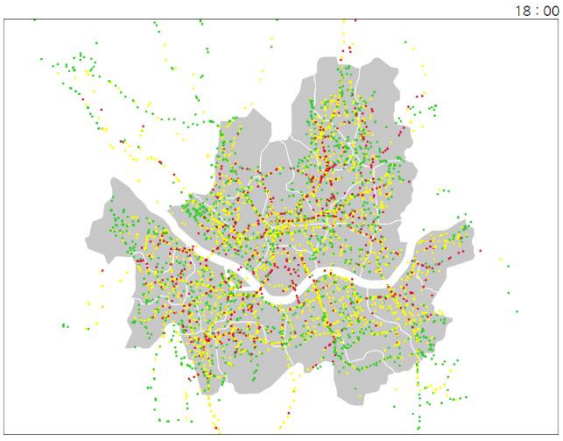
상관관계의 분석 결과, 직접 구해준 버스혼잡도 데이터는 어느 정도 신뢰할 수 있는 데이터라는 사실을 알 수 있게 되었고, 서로 상관관계가 있을 것이라고 예상했던 자치구별 통학·통근 인원, 자치구별 세대수와 대중교통은 연관관계가 거의 없다는 것을 알 수 있었으며, 상가 또는 기업 수와 대중교통과의 관계는 직접 구해보지는 못했지만 어느 정도 관련이 있을 수 있을 것이라고 예상을 하였다. 아래의 그림은 어느 정도 상관관계를 보였던 버스와 다른 요소들의 상관관계를 정리하여 나타낸 것이다.



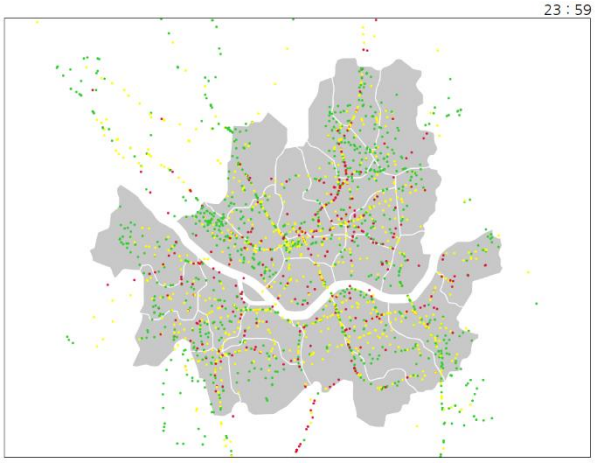
결론 및 기대효과

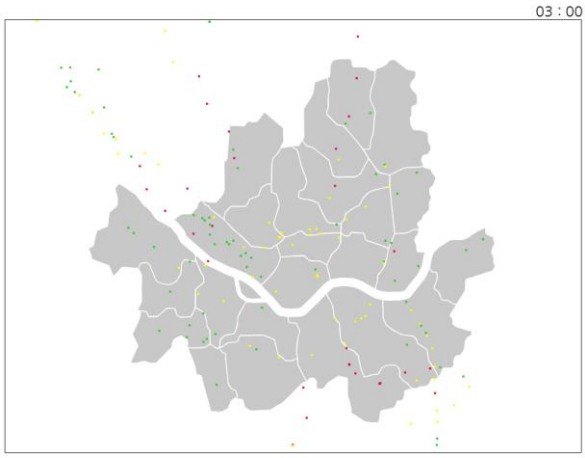
먼저, 주어진 데이터들을 가공하여 버스의 혼잡도 데이터를 만들었다. 이 데이터를 통해서는 버스 한 대 당 탑승인원을 시간별로 알 수 있으며, 이를 동적으로 시각화해주어서 시간별로 버스의 이동 경로와 그 버스의 혼잡도를 알 수 있다. 기존의 데이터들은 버스정류장에서 특정 시간대에 그 버스에 몇 명이 탔는지에 대한 정보만 제공해준다. 그래서 이 정보만을 가지고 있을 때는 통계적으로 내가 어떤 시간대에 버스를 타면 그 버스에 몇 명이 탈지에 대한 정보는 알아내기가 힘들고, 이에 대해서 실시간으로 탑승객들의 GPS정보를 받아서 혼잡도를 나타내어주는 어플은 있지만 통계적으로 버스의 혼잡도를 나타내어주는 데이터는 찾기가 힘들었다. 이 데이터를 사용하면 개인의 입장에서는 자신이 탈 버스가 그 시간대에 사람이 많이 타고 있을지에 대한 정보를 알 수 있고, 정책적인 측면에서도 버스들의 전체적인 혼잡도를 알 수 있기 때문에 여러가지 제안을 할 수 있을 것이다. 아래에 나올 그래프들은 동적 구현된 버스 혼잡도를 나타내는 그래프이며 특정 시간대의 버스 혼잡도의 모습을 보여준다. 그래프에서의 작은 원 하나가 한 대의 버스를 의미하며, 이 원이 초록색이면 버스 탑승인원이 상위 50% ~ 상위 100%, 노란색이면 상위 10% ~ 상위 50%, 빨간색이면 상위 10% 이상이라고 할 수 있다. 그래프를 통해 얻은 새로운 관점들에 대해서 몇가지 예시를 들면 다음과 같다.



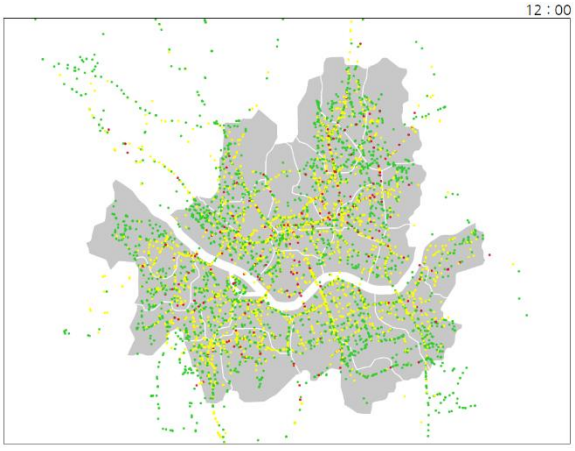


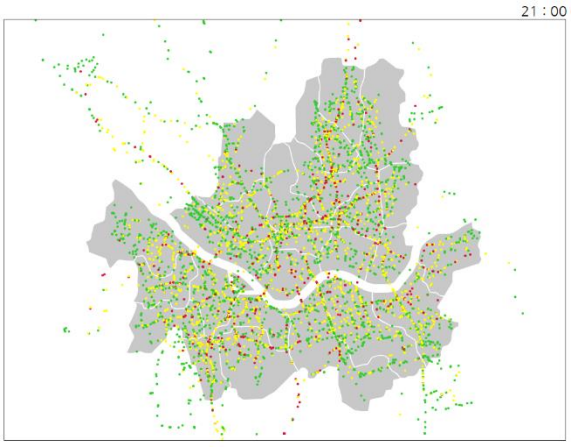
첫번째로, 출근시간과 퇴근시간의 그래프를 보면 서울 전체적으로 상당히 혼잡한 모습을 보이는 것을 알 수 있다. 그리고 이 그래프에서 서울의 남동쪽에 해당되는 서초구와 강남구의 남쪽부분을 보면, 이 지역에는 버스가 많이 다니지 않는 것을 알 수 있다. 다른 지역에 버스가 많이 없는 곳은 대부분 산이나 강이 있기 때문인데, 이 지역 같은 경우는 최근에 내곡 공동주택지구, 세곡 2 공공 주택지구, 구룡도시 개발구역 등 새로운 주거단지가 조성되고 있거나 조성이 된 지역이다. 아직 인구가 그렇게 많지는 않기 때문에 혼잡도가 매우 높게 나타나지는 않지만 이 지역 역시 주거단지가 들어오고 있는 지역이기 때문에 이 지역에 대해서는 버스 노선의 증설이 필요하다고 할 수 있다.





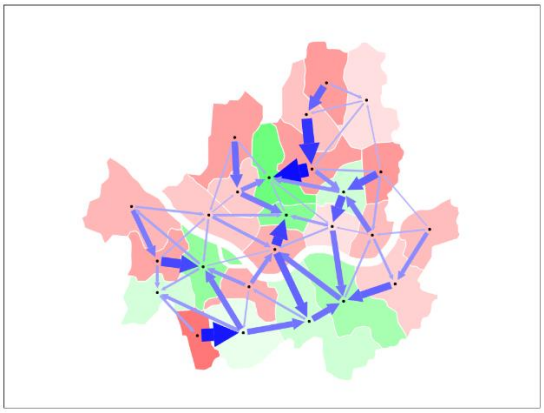
다음으로는 24시 직전과 막차들이 거의 운행을 종료하는 시점인 오전 3시의 그래프를 보면, 일단 24시에 상당히 많은 버스들이 매우 높은 혼잡도를 보이고 있다는 것을 알 수 있고, 오전 3시에는 버스가 몇 대 다니지 않지만 그 중에서도 매우 높은 혼잡도를 보이는 버스들이 운행을 하는 것을 알 수 있다. 이를 통해서, 서울시에 심야버스에 대한 수요가 많다는 것을 추측할 수 있으며, 새벽 시간까지 꾸준하게 혼잡도가 높은 구간들에 대해서는 비슷한 노선의 심야버스를 만들거나 막차 시간을 조금 더 늦추는 등의 방법을 제안할 수 있다. 이에 대해서는 실제로 서울시에서 서울 지하철 24시간 운행을 준비하고 있으며, 연말을 대비하여 N854번과 N876번의 심야버스 노선을 신설하는 등 이에 대한 대책을 마련하고 있다.



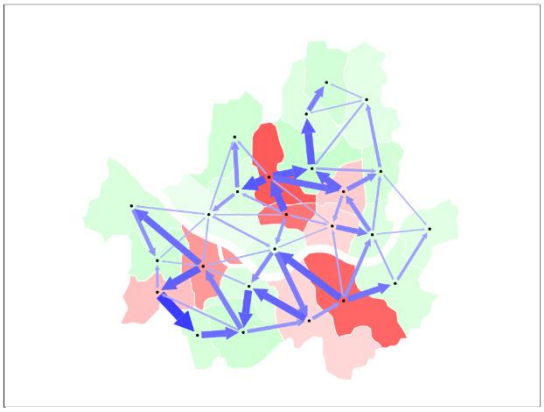


마지막으로는 특정 버스 노선의 시간대별 수요를 보기 위해 서울에서 경기도 쪽으로 가는 버스의 이동을 12시와 21시 데이터를 기준으로 비교를 해보도록 하겠다. 일단, 그래프들을 보면 북서쪽, 경기도 파주시 쪽으로 뻗어가는 버스 노선을 볼 수 있을 것이다. 이 노선의 경우는 거의 모든 시간 원활한 혼잡도를 나타내고 있다. 즉, 이러한 노선들에 대해서는 버스 배차간격을 더 늘리는 것이 버스를 운영하는 입장에서 더 효율적일 것이라고 예측할 수 있다. 또 이번에는 서울의 남쪽인 과천 쪽으로 가는 버스 노선을 보면, 21시에 특별히 매우 혼잡하다는 것을 알 수 있다. 이와 같은 정보가 있다면 버스의 배차간격을 항상 10분과 같이 계속 일정하게 하는 것이 아니라 여유 있을 때는 15분, 혼잡할 때는 5분과 같이 탄력적으로 버스의 배차를 바꾸어 줄 수도 있을 것이다.
지금까지 살펴본 바와 같이 혼잡도 데이터를 이용하면 개인 뿐만 아니라 버스를 운영하는 입장, 그리고 정책적인 입장에서도 기존보다 훨씬 더 확장된 시야로 데이터를 볼 수 있을 것으로 기대된다.

다음으로는 우리가 가공해준 데이터를 바탕으로 자치구별 이동인원을 구해보았다. 이 이동인원이 버스 이용자만을 반영하고 있는 이동인원이지만, 이 인원자체로도 충분히 가치가 있다고 할 수 있을 것이다. 일단, 기존의 데이터들, 특히 공개된 데이터들 중에서는 일간 자치구별 이동인원을 나타내는 데이터는 찾아보기 힘들었다. 대부분의 유동인구 데이터는 거주지를 기준으로 추산된 데이터 이었으며, 그나마 비슷하다고 할 수 있는 통학·통근 데이터와 같은 데이터 역시 거주지를 기준으로 한 설문조사에 바탕을 둔 데이터이고 이 역시 출퇴근시간의 이동경향만을 대략적으로 파악할 수 있는 정도였다. 하지만, 우리가 구해준 데이터와 구현한 시각화 자료를 통해 일간 원하는 시간대에 어느 구에서 어느 구로 사람이 얼마나 이동하는지를 통계적으로 파악할 수 있게 되었다. 이 정보는 일단 먼저 구해준 버스 혼잡도 데이터에 기본을 하고 있고, 그 혼잡도 데이터는 정부에서 지원하는 가공하기 전의 승·하차 인원 데이터와 큰 상관관계를 나타낸다는 것을 확인하였으므로, 어느 정도 유의미한 데이터가 될 것 이라고 추측할 수 있다. 아래의 그래프는 서울시 지도에 각각 출퇴근시간별로 자치구 간 인구의 이동을 나타낸 그래프이다. 구 지역이 초록색으로 되어 있는 부분은 인구의 유입이 많은 자치구, 빨간색으로 되어 있는 부분은 인구의 유출이 많은 자치구이며, 파란색 화살표는 구간 인구의 이동을 나타낸다.



오전 8시의 구간 이동 그래프



오후 7시의 구간 이동 그래프

이 데이터가 유의미한 정보인지를 파악하기 위해서 그나마 연관성이 있다고 판단되었던 통학·통근 데이터와 출퇴근시간의 데이터와 비교를 해보았다. 통학·통근 인원이 가장 많은 상위 3개의 자치구인 송파구, 강서구, 노원구 모두 출근시간(08:00)에는 인구의 유출이 많고, 퇴근시간(19:00)에는 인구의 유입이 많은 것을 확인할 수 있었다. 하지만, 이 자치구들이 출근시간의 유출과 퇴근시간의 유입이 가장 많은 자치구는 아니었는데 이 이유에 대해서 생각해보면 다음과 같다. 위의 세 개의 구는 자치구별 통학·통근 데이터를 보면 같은 자치구내의 통학·통근 인원 역시 많은 자치구에 속했다. 즉, 많은 사람들이 그 자치구에서 통근 또는 통학을 하지만 그만큼 많은 사람들이 그 자치구로 학업 또는 일을 하러 올 수 있다고 할 수 있다. 다음으로는, 우리가 구해준 자치구별 이동인원은 출발지와 도착지를 비교해준 것이 아니고 버스가 자치구를 이동할 때 그 버스에 타고 있는 승객의 수를 기준으로 구해준 것이기 때문에 도착지까지 갈 때 다른 자치구를 거쳐서 가는 것 역시 포함이 된다는 점도 있다. 또한, 주요 상업단지와 대기업들이 모여 있는 자치구인 강남구, 종로구, 영등포구를 살펴보면 뚜렷하게 출근시간에는 다른 자치구들에 비해 인구 유입이 매우 많이 일어나고, 퇴근시간에는 인구 유출이 많이 일어난다는 것을 확인할 수 있다. 이와 같은 정보들을 통해서 자치구별 이동인구 역시 어느 정도 신뢰할 수 있는 정보를 제공한다는 것을 확

인할 수 있다. 이 데이터는 그간 제공되던 데이터에 비해 훨씬 짧은 주기의 인구이동 경향을 나타내기 때문에 지하철역 증설, 새로운 버스 노선 추가 또는 정류장의 증설 등 다양한 교통 기획 또는 서울 내 인구 밀집 방지를 위한 정책 수립 등 다양한 분야의 참고자료로 쓰일 수 있을 것으로 기대된다.

아쉬운 점

이번 프로젝트를 하면서 몇가지 아쉬운 점이 있어서 이 부분에 대해서 적고 보고서를 마무리하고자 한다. 첫번째 아쉬운 점은 버스 분석의 경우 데이터가 한정되어 있어서 시간이나 거리를 가정하여 구했기 때문에 혼잡도를 구하는 과정 등에서 정확한 분석이 어려웠다는 것이다. 우리가 사용한 데이터의 경우는 교통카드 로그 정보였기 때문에, 이 정보의 raw data를 구하면 카드 ID정보가 있기 때문에 사용자에 따라 정확하게 승차 정류장과 하차 정류장을 구할 수 있을 것이라고 예상이 되며, 이동 거리 및 시간 역시 버스의 운행정보가 있어서 이 정보를 이용할 수 있다면 더욱 더 정확한 분석을 할 수 있었을 것 같다고 생각한다.
두번째 아쉬운 점은 상관관계를 분석할 때 대중교통의 승·하차 인원이 상가/대기업과 상관관계가 클 것으로 예상이 되었는데 이와 관련된 데이터의 부족으로 이를 확인해보지 못한 점이 아쉬움이 남는다. 마지막으로는 지하철의 혼잡도를 구하지 못한 부분인데, 일단 지하철의 경우는 승·하차 인원 정보만으로 그 인원이 상행선을 탔는지 하행선을 탔는지를 알 수 없었고, 지하철은 환승도 상당히 많이 일어날 것으로 예상이 되는데 환승을 할 때는 카드의 로그 기록이 남지 않기때문에 환승 정보에 대해서 알 수 없다는 분석의 한계점이 있었다. 또한, 승·하차 및 혼잡도에 대한 데이터를 찾아볼 수는 있었지만, 서울교통공사에서 운영하는 1~8호선까지만 데이터가 있었고, 과거의 데이터의 경우는 서울교통공사와 서울메트로가 합병이 되면서 기존 서울메트로가 제공을 하던 데이터들이 사라져서 데이터 수집에 어려운 부분이 있었다.