자유주제
데이터 분석

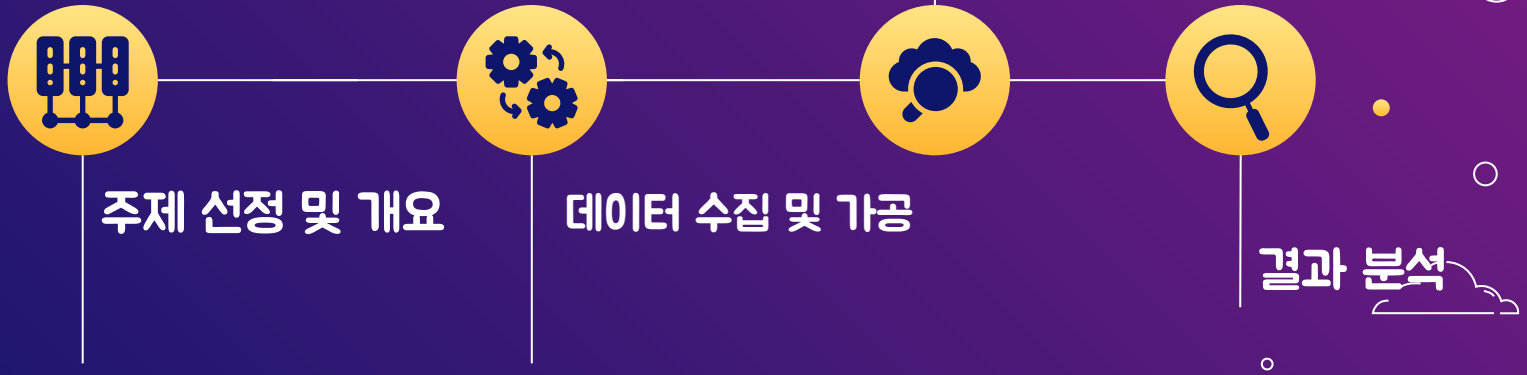
DATA ANALYSIS STUDY

TEAM 3

유지원, 송지원, 김관엽

TOPIC

코로나 전/후 서울 지하철 7호선의 '역별 승하차 인원 감소율'을 비교하여 코로나의 타격 정도와 그 이유 분석



01 주제 선정 및 개요

주제 논의

- 자료 idea ➡ http://www.seoulmetro.co.kr/kr/board.do?menuIdx=548
- 서울교통공사에서 제공하는 지하철 수송실적 데이터 이용
- 통학러가 가장 많은 학교는??(인원 분석)
- 앉아서 지하철을 타려면 어느 시간에 타야할까??(시간대별 분석)
- 코로나로 타격이 가장 큰 지하철 역은??(감소율 분석)

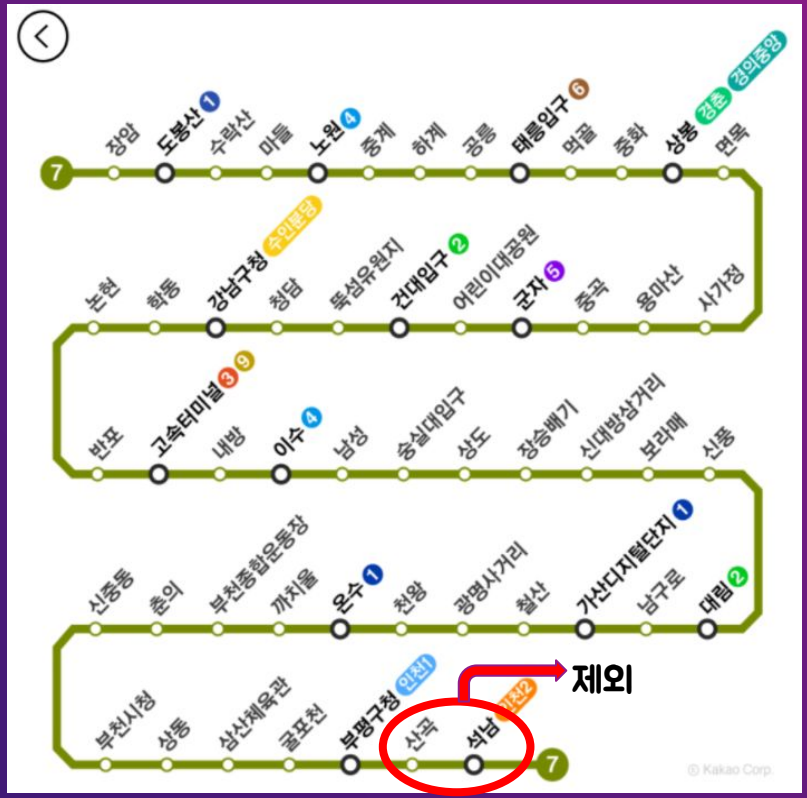
주제 선정

코로나 전/후 서울 지하철 7호선의 역별 승하차 인원 감소율을 비교하여 코로나의 타격 정도와 그 이유 분석

7호선은 세종대를 포함한 다수의 대학을 지나가고, 노원구 등의 주거지역과 강남, 구로디지털단지 등의 상업지역, 그리고 뚝섬유원지와 같은 관광지역을 모두 관통하는 노선인만큼 다양하고 폭 넓은 분석이 가능함

총 53개 역 중 산곡역과 석남역은 2021년 5월에 개통하여 코로나 이전 데이터 없음>>분석에서 제외

분석 대상은 장암역~부평구청역으로 총 51개 역



단계별 개요

논의사항 1. 데이터 수집

-코로나 시작 시점은?
ㄴ 코로나로 거리두기 정책을 시행하기 시작한 2020년 4월을 기준

-비교 대상은?
ㄴ코로나 이전 18개월(18.04~19.11)VS코로나 이후 18개월(20.04~21.11)
ㄴ월별 차이로 인한 혼선을 줄이기 위해 연도만 다르게 함

-주어진 자료 외에 미리 조사해야 할 것은?
ㄴ대학교 등 타격이 큰 시설과 붙어있는 역 조사
ㄴ주거지역, 상업지역, 관광지역 등과 붙어있는 역 조사

논의사항 2. 데이터 가공

-역별 이용 인원 계산 기준은?
ㄴ실질적으로 역을 이용하는 승하차 인원만 고려
ㄴ환승 유입 인원은 고려하지 않음

-시각화에 필요한 주된 데이터는?
ㄴ단순 인원 계산은 역 이용객에 따라 큰 차이가 나므로 비교 불가
ㄴ즉 비율을 이용한 감소율 사용
ㄴ승하차 감소율=(코로나 이전 승객-코로나 이후 승객)/(코로나 이전 승객)*100(%)

논의사항 3. 데이터 시각화

-차트 시각화
ㄴ승하차 감소율이 가장 높은 역 5개
ㄴ승하차 감소율이 가장 낮은 역 3개

-구글 맵스(Google Maps)
ㄴ총 51개 역 모두를 지도에 표시
ㄴ감소율이 높으면 빨간색, 낮으면 파란색으로 표시
ㄴ지역별 현황을 한눈에 비교 가능

논의사항 4. 결론 분석

-실제 결과값을 도출하기 전 미리 예측을 할 것임

-다음 사항들이 미리 예측되는 결과이다.
ㄴ역 주변시설이 무엇이냐에 따라 결과값이 달라질 것
ㄴ대학교가 붙어있는 역은 비대면의 영향으로 감소율이 매우 클 것이다.
ㄴ주거지역과 붙어있는 역은 다른 지역에 비해 상대적으로 코로나로 인한 영향이 작으므로 감소율이 작을 것이다.

-실제 분석에서는 기사 등 다양한 매체도 참고할 예정

02 데이터 수집 및 가공

데이터 수집

-서울교통공사에서 제공하는 '월별 수송 실적 데이터'를 활용
-승차인원과 하차인원을 더한 수송인원을 선택

승차인원(a) | 하차인원(b) | 승하차인원(a+b) | 환승유입인원(c) | 수송인원(a+c) | 권종별 승차인원

-data1: 코로나 이전 18개월(18.04~19.11),
data2: 코로나 이후 18개월(20.04~21.11) 의 데이터만 excel파일로 정리

	A	B	C	D	E	F
1	역명	4월	5월	6월	7월	8월
2	장암	109,898	117,564	113,921	123,165	111,818
3	도봉산(7)	675,493	702,991	644,143	619,496	599,197
4	수락산	851,948	889,870	844,877	850,777	810,864
5	마들	728,056	751,631	703,912	719,060	696,706
6	노원(7)	1,329,066	1,398,591	1,305,919	1,377,088	1,350,569
7	중계	965,410	990,088	917,528	931,102	892,096

	A	B	C	D	E	F
1	역 명	4월	5월	6월	7월	8월
2	장암	58,846	63,815	66,181	70,215	59,466
3	도봉산(7)	220,328	242,200	251,082	260,291	214,582
4	수락산	309,941	340,735	348,420	363,458	310,439
5	마들	275,194	300,489	316,540	333,420	286,255
6	노원(7)	455,755	511,893	525,438	561,648	478,063
7	중계	325,859	361,457	384,107	398,101	334,060

데이터 가공

(1) 코로나 후 가장 승객 감소율이 큰 역 5개 추출하기
(2) 코로나 후 가장 승객 감소율이 작은 역 3개 추출하기

-> 각 역의 감소율을 구한 뒤 오름차순,내림차순으로 정리하기

(3) 코로나 전, 코로나 후의 월 평균 승객수 구하기

-> 평균값 계산해서 추가하기

데이터 가공

(1) Data1 Excel 파일 불러오기

1.필요한 모듈 import

```
import pandas as pd
```

2. read_excel()

```
pre_covid_subway=pd.read_excel('/content/Data/data1.xlsx')
```

3. head() , 파일을 잘 불러왔는지 확인

```
pre_covid_subway.head()
```

	역명	4월	5월	6월	7월	8월	9월	10월	11월	12월
0	장암	109898	117564	113921	123165	111818	105277	113700	107820	98481
1	도봉산(7)	675493	702991	644143	619496	599197	625404	707077	673147	621491
2	수락산	851948	889870	844877	850777	810864	798146	880083	862475	844261
3	마들	728056	751631	703912	719060	696706	673518	751304	745782	718543
4	노원(7)	1329066	1398591	1305919	1377088	1350569	1242644	1379548	1360951	1369822

데이터 추출

(2) 승객수의 합계, 평균 구하기

1. 합계 구하기

```
pre_covid_subway['합계']=pre_covid_subway['4월']+pre_covid_subway['5월']+pre_covid_subway['6월']+p
```

2. 잘구해졌는지 확인하기: head()

```
post_covid_subway.sort_values(by='합계', ascending=False).head()
```

	역명	합계
37	가산디지털단지(7)	47732990
39	광명사거리	32028678
38	철산	29623835
4	노원(7)	26886710
22	학동	26477754

2. 평균 구하기

```
pre_covid_subway['평균']=pre_covid_subway['합계']//18
```

데이터 추출

(3) 승객 감소율 구하기

1. 감소인원 합계 구하기 : (코로나 전 승객수 합계) - (코로나 후 승객수 합계)

```
pre_covid_subway['감소인원합계']=pre_covid_subway['합계']-post_covid_subway['합계']
```

2. 감소율 구하기

```
pre_covid_subway['감소율']=pre_covid_subway['감소인원합계']/pre_covid_subway['합계']*100
```

데이터 추출

(4) 코로나 후 가장 승객 감소율이 큰 역 5개 추출

: 건대입구, 어린이대공원(세종대), 숭실대입구, 대림, 고속터미널

```
pre_covid_subway.sort_values(by='감소율', ascending=False).head()
```

	역명	4월	5월	6월	7월	8월	9월	10월	11월	12월	1월	합계	평균	감소인원 합계	감소율
18	건대입구(7)	1062624	1110224	1001350	1019705	1025240	965090	1094820	1196538	1150958	1017026	20928062	1162670	14633848	69.924525
17	어린이대공원	1189023	1287507	995391	801911	819649	977168	1120165	1064740	923824	767953	19827926	1101551	13719745	69.194050
29	숭실대입구	1068978	1079721	940768	801577	780824	936054	1052822	1064684	937519	825791	18998303	1055461	13016729	68.515219
35	대림(7)	654678	669806	633984	654245	663946	678474	698674	671892	694718	662250	13207851	733769	8961203	67.847548
25	고속터미널(7)	1228278	1275509	1170050	1198954	1190759	1150768	1222003	1237218	1242088	1200218	24006414	1333689	15927055	66.344998

데이터 추출

(5) 코로나 후 가장 승객 감소율이 작은 역 3개 추출
: 장암, 천왕, 군자

```
pre_covid_subway.sort_values(by='감소율', ascending=True).head(3)
```

	역명	4월	5월	6월	7월	8월	9월	10월	11월	12월	1월	2월	합계	평균	감소인원합계	감소율
0	장암	109898	117564	113921	123165	111818	105277	113700	107820	98481	96454	82553	2186614	121478	968754	44.303841
40	천왕	534644	539655	512809	525390	504325	488816	554545	549544	531422	531386	456341	10968258	609347	5583429	50.905340
16	군자(7)	762649	811180	781148	806521	785798	736041	831717	829835	818346	825249	698249	16094036	894113	8741349	54.314213

데이터 추출

(6) 월 평균 지하철 이용객 추출

-코로나 전

```
pre_covid_subway.iloc[:,[0,21,23,25]].to_excel('pre_average_address.xlsx')
pre_average_address = pd.read_excel('/content/pre_average_address.xlsx')
pre_average_address.head()
```

	Unnamed: 0	역명	주소	평균	감소율
0	0	장암	경기도 의정부시 동일로 121	121478	44.303841
1	1	도봉산(7)	서울특별시 도봉구 도봉로 964-40	723892	63.900611
2	2	수락산	서울특별시 노원구 동일로 1662	934328	60.573803
3	3	마들	서울특별시 노원구 동일로 1530-1	803306	57.226845
4	4	노원(7)	서울특별시 노원구 상계로 69-1	1493706	63.348004

데이터 추출

(7) 월 평균 지하철 이용객 추출

-코로나 후

```
post_covid_subway.iloc[:,[0,21,23]].to_excel('post_average_address.xlsx')
post_average_address = pd.read_excel('/content/post_average_address.xlsx')
post_average_address.head()
```

	Unnamed: 0	역명	주소	평균
0	0	장암	경기도 의정부시 동일로 121	67658
1	1	도봉산(7)	서울특별시 도봉구 도봉로 964-40	261320
2	2	수락산	서울특별시 노원구 동일로 1662	368370
3	3	마들	서울특별시 노원구 동일로 1530-1	343599
4	4	노원(7)	서울특별시 노원구 상계로 69-1	547473

03

데이터 시각화

: 지하철역을 지도에 표시하기

(1) 승객 수 감소율이 가장 큰/작은 역 비교하기

1. 필요한 모듈 import

```
import seaborn as sns
import matplotlib.pyplot as plt
```

- Seaborn
- Matplotlib

Seaborn, matplotlib -> 한글 입력 시 깨짐
데이터 분류에 한글이 들어가므로 한글 폰트 추가함

```
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf
```

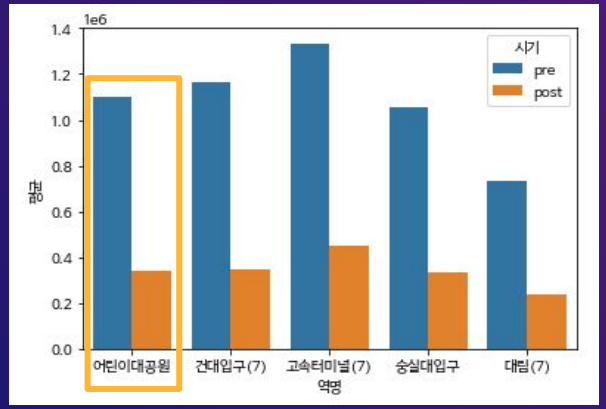
```
plt.rc('font', family='NanumBarunGothic')
```

(1) 승객 수 감소율이 가장 큰/작은 역 비교하기

2. Barplot 그리기

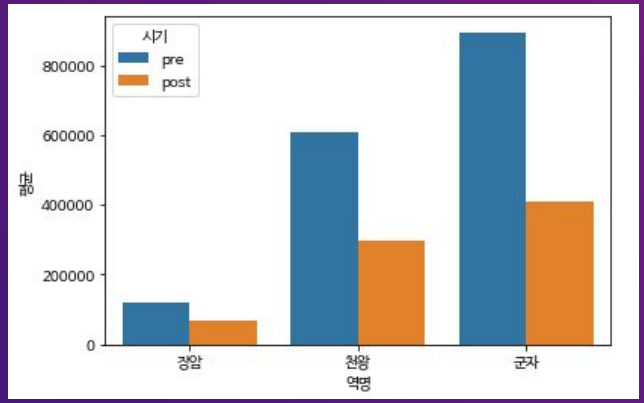
a. 감소율 가장 큰 5개 역

```
sns.barplot(x='역명',y='평균',hue='시기',data=df_max)
```



b. 감소율 가장 작은 5개 역

```
sns.barplot(x='역명',y='평균',hue='시기',data=df_min)
```



(2) 지하철 승객 수 감소율 googlemaps에 나타내기

1. 필요한 모듈 import

```
!pip install googlemaps

import folium
import googlemaps
```

- Folium
- Googlemaps

2. API 키로 googlemaps 읽어오기

```
gmaps_key='                                                    '
gmaps = googlemaps.Client(key=gmaps_key)
```

(2) 지하철 승객 수 감소율 googlemaps에 나타내기

3. (googlemaps) 위도, 경도 정보 받아오기

```
lat = []
lng = []

for n in pre_average_address.index:
        target_name = pre_average_address['주소'][n]
        gmaps_output = gmaps.geocode(target_name)
        location_output = gmaps_output[0].get('geometry')
        lat.append(location_output['location']['lat'])
        lng.append(location_output['location']['lng'])



pre_average_address['lat'] = lat
pre_average_address['lng'] = lng
pre_average_address.head()
```

지오코딩(geocoding)

- 주소를 지리적 좌표로 변환하는 것

배열에 저장

	Unnamed: 0	역명	주소	평균	감소율	lat	lng
0	0	장암	경기도 의정부시 동일로 121	121478	44.303841	37.700080	127.053120
1	1	도봉산(7)	서울특별시 도봉구 도봉로 964-40	723892	63.900611	37.689080	127.046509
2	2	수락산	서울특별시 노원구 동일로 1662	934328	60.573803	37.677155	127.055307
3	3	마들	서울특별시 노원구 동일로 1530-1	803306	57.226845	37.665027	127.058007
4	4	노원(7)	서울특별시 노원구 상계로 69-1	1493706	63.348004	37.655867	127.061991

(2) 지하철 승객 수 감소율 googlemaps에 나타내기

4. (folium) marker로 지도에 지하철 역 표시

```
map = folium.Map(location=[37.6049559175497, 127.02],zoom_start=11.3)

maxs=["건대입구(7)","어린이대공원","숭실대입구","대림(7)","고속터미널(7)"]
```

center 위치 설정해 지도 그리기

승객 수가 가장 크게 감소한 5개 역을 maxs 배열에 저장

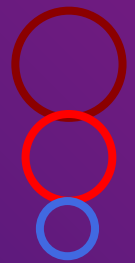
```
for n in pre_average_address.index:
  if pre_average_address['감소율'][n]>pre_average_address['감소율'].mean():
    if pre_average_address['역명'][n] in maxs:
          folium.CircleMarker([pre_average_address['lat'][n],pre_average_address['lng'][n]],
          radius=20,color = ' #8B0000 ',
          fill_color='#8B0000').add_to(map)
    else:
      folium.CircleMarker([pre_average_address['lat'][n],pre_average_address['lng'][n]],
                    radius=15,color = ' #FF0000 ',
                    fill_color='  #FF0000 ').add_to(map)
  else:
    folium.CircleMarker([pre_average_address['lat'][n],pre_average_address['lng'][n]],
                    radius=8,color = '  #4169E1',
                    fill_color='  #4169E1').add_to(map)
```

For 반복문으로 각 역의 감소율 조회

If 해당 역의 감소율 > 감소율 평균 :

- If 해당 역 in maxs[] :
- Else :

Else 해당 역의 감소율 < 감소율 평균 :



(2) 지하철 승객 수 감소율 googlemaps에 나타내기

4. (folium) marker로 지도에 지하철 역 표시

감소율 Top 5로 큰 역

감소율이 평균치보다 큰 역

감소율이 평균치보다 작은 역

04 결과 분석

결과 분석

1) 감소율이 가장 높은 Top 5 Station

-건대입구역, 어린이대공원역, 숭실대입구역(각 1~3위)

ㄴ예상대로 대학교와 인접해 있는 역 3 곳이 Top 1~3위를 차지하였다.

결과 분석

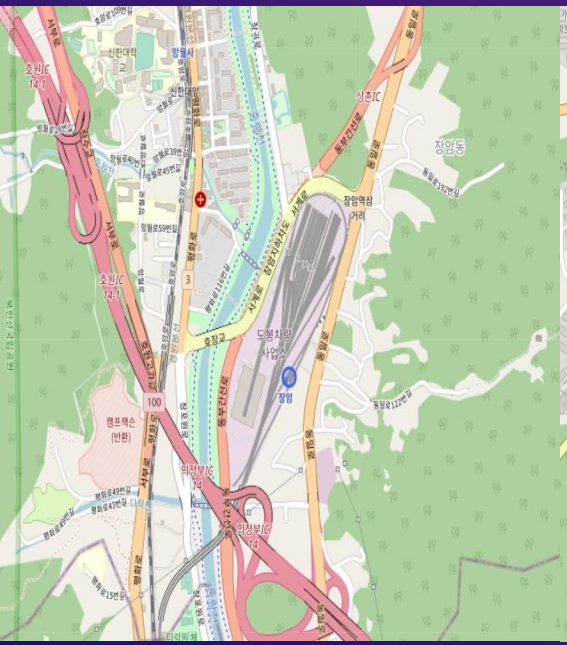
-고속터미널역(5위)
ㄴ코로나로 인해 지역 간 이동이 줄어들었음을 확인해주는 지표이다.

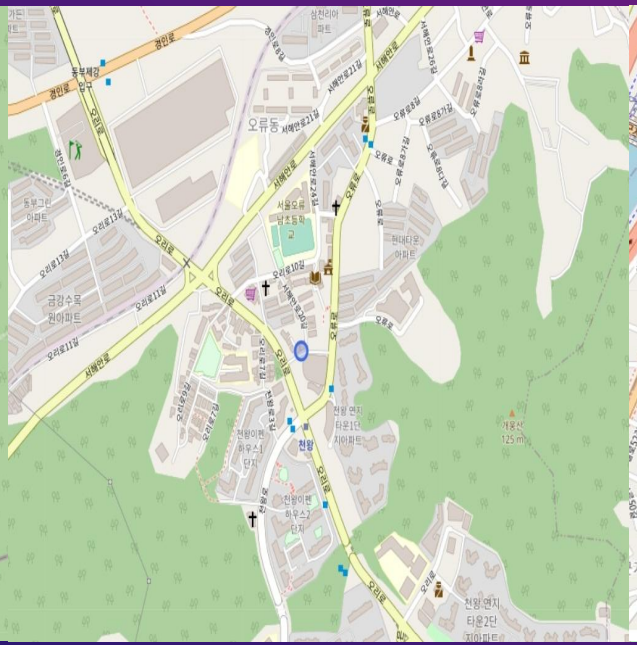
-대림역(4위)
ㄴ구직 인구가 많은 지역으로, 코로나로 인해 위축되었다고 예상할 수 있다.

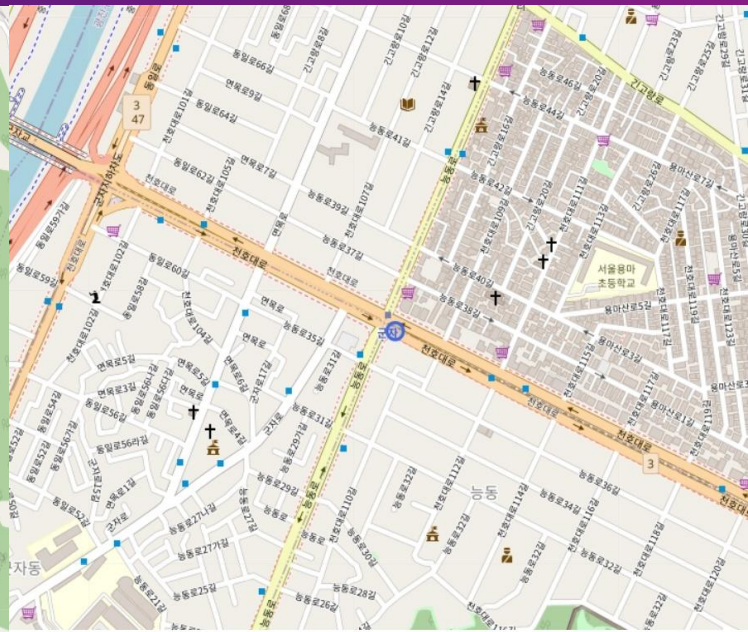
결과 분석

2) 감소율이 가장 낮은 Top 3 Station

-장암역, 천왕역, 군자역(각 1~3위)







결과 분석


: 유동인구가 적은 주거 지역
: 유동인구가 많은 주거 지역
: 상업 지역
: 관광 지역
노원구
중랑구
강남
뚝섬유원지
부천 동부, 구로구
부천 서부, 부평
신길동, 신대방동
구로디지털단지

결과 분석

3) 기타 지역

-유동인구가 많은 주거 지역(노원구, 부천 서부, 부평 등)
ㄴ원래 이동량이 많은 지역이었으므로 감소율 큼

-유동인구가 적은 주거 지역(중랑구, 부천 동부, 구로구, 신길동, 신대방동 등)
ㄴ원래 이동량 적음+상대적으로 코로나 영향력이 적은 주거 지역=감소율 작음

-상업지역(강남, 구로디지털단지)
ㄴ주요 시설이 많이 몰려있어 코로나에도 불구하고 감소율 작음

-관광지역(뚝섬유원지)
ㄴ코로나로 인해 관광객이 줄었으므로 감소율 큼

예측한 결과vs실제 결과

1) 역 주변시설이 무엇이냐에 따라 결과값이 달라질 것
ㄴ실제로도 역 주변 환경에 따라 감소율이 다름.

2) 대학교가 붙어있는 역은 비대면의 영향으로 감소율이 매우 클 것이다.
ㄴ감소율이 가장 큰 역 1~3위가 모두 대학교가 붙어 있는 역이므로,
정확하게 예측함.

3) 주거지역과 붙어있는 역은 다른 지역에 비해 상대적으로 코로나로 인한
영향이 작으므로 감소율이 작을 것이다.
ㄴ주거 지역에 포함되어 있더라도, 원래 유동인구가 많았던 지역은 감소율이
크게 나타난다는 사실을 새롭게 확인할 수 있었음.

느낀 점

유지원

생각보다 지하철 이용객이 엄청나게 줄어서 매우 놀랐고 주변 사람들에게도 이 데이터를 보여주니 다들 놀랍다는 반응이었다. 팀원들과 여러가지 주제를 던져보는 시간이 재미있었고 생각만 했던 것을 데이터를 가져와 조작하고 수치로 확인하는 과정이 매우 흥미로웠다. 특히 데이터를 두고 그 결과가 도출된 원인을 추론하는 것이 특히 재미있었다.
데이콘, 캐글 등 공공데이터를 얻을 수 있는 곳이 상당히 많고 그 커뮤니티에서 다양한 시도들이 이뤄지고 있는 것이 신기했다.
혼자 했으면 갖지 못했을 다각도의 시선으로 데이터를 분석할 수 있는 아주 좋은 경험이었다.

송지원

주제선정 과정에서 여러가지를 이야기하면서 구체화하는 과정이 재밌었다. 데이터 분석을 하기 전 예측했던 것들과 분석 결과가 어느정도 일치하는 것을 보고 신기했던 것 같습니다. 또 실제로 코로나 전과 후의 수치를 비교해보니 생각보다 차이가 심해서 놀랐습니다. 책을 따라 공부하다가 조원들과 함께 주제 선정부터 데이터 가공, 시각화 방법 등을 고민하고 직접 해보니 정말 무언가를 얻어갈 수 있는 스터디 활동이었다고 생각이 듭니다.

김관엽

책에 나와있는 내용을 그대로 따라가는 것이 아닌, 주제 선정 부터 기획, 데이터 수집, 가공, 시각화, 결과 분석 까지 모두 조원분들과 함께 새롭게 구성하여 해보는 게 굉장히 재미있었다. 시대적 상황인 코로나를 반영하여 흥미롭고 획기적인 주제를 잡은 것, 그리고 그것에 딱 걸맞는 데이터 자료가 있어서 매우 만족스러웠다. 또한 단순히 결과값 도출로만 끝나는 것이 아닌, 세세한 분석을 통해 미리 예측했던 결과와 비교해 보는 것도 정말 재미있었다. 이번 스터디로 또 한번 상당히 유익한 경험을 쌓을 수 있어서 너무 좋았다.
함께 열심히 참여해 주신 조원분들께도 정말 감사드린다.

감사합니다.