극데노
(극한의 데이터 노예)
유제우
이서영
이현중

SEOUL BIG DATA CAMPUS MENTORING

Part 1

목차

Part 1

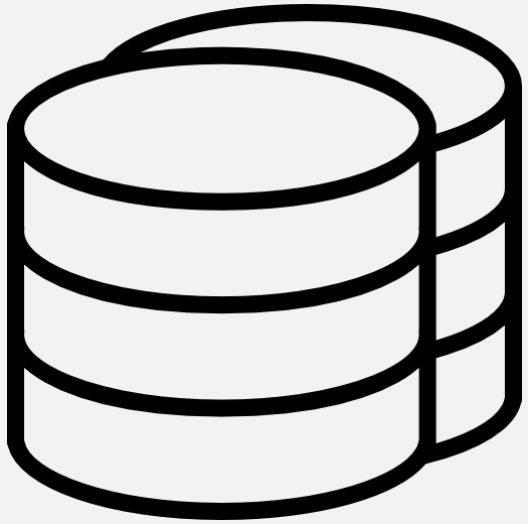
데이터 선정

Part 1

데이터 선정

Dataset

- 30분 지하철 30분 단위 이용 통계
- 서울시 지하철 시간대별 승객 수
- 우수중소기업 공간데이터
- 수도권지하철 공간데이터
- 서울시 주요시설과 집객 시설 공간데이터



눌러주세요

Part 2

주제

Part 2

주제

서울시 시민의 지하철, 대중교통 데이터, 30분 단위 이용 통계, 여러 서울시 공간데이터를 활용해 사람들의 출발지와 도착지 정보를 이용하여 시간별 혼잡 상황을 파악하고, 혼잡도를 미리 예측한다. 그 후, 몰리는 특정 역의 이유를 찾고 사람들의 출발역, 도착역 정보를 파악하여 균등한 객차 혼잡비율을 조절해 시각화를 통한 인사이트를 도출한다.

지하철을 이용하는 인원은 항상 일정하지 않다. '지옥철' 이라는 말이 있듯 특정 시간, 특정 요일에 그리고 특정 역에 집중되는 것은 누구나가 체감한 통념이다. 우리는 이런 통념을 구체적인 수치로 파악함으로써 특정 시간, 특정 역에 이용자가 집중된다는 것을 증명하고, 이를 토대로 집중되는 요일, 시간, 역을 알고자 한다.

Part 2

데이터 설명

20년도 승하차 데이터

```
# value값에 쓰잘데기없는 `기호 없애기

subway20 = subway20.apply(lambda x: x.str.strip("`"), axis = 0)
subway20.head(3)
```

	YEAR	MONTH	DAY	HOUR	HALF_HOUR	LINE_NM	STATION_NM	GETON_CNT	GETOFF_CNT
0	2020	01	01	04	30	1호선	서울역	11	1
1	2020	01	01	04	30	1호선	시청	1	0
2	2020	01	01	04	30	1호선	종각	5	2

Part 2

데이터 설명

서울시 지하철 시간대별 승객수

```
# 서울시 지하철 시간대별 승객수 19년
SEOUL19 = DBF('data/서울시 지하철 시간대별 승객 수/TB_SEOUL_TRAIN_PASSN_2019.dbf')
SEOUL19 = pd.DataFrame(SEOUL19)
SEOUL19.head(3)
```

	ID	STN_IDN	TGRP_CODE	STN_NM	BB_RT	US_YY_DD	PS_00TM	AG_00TM	PS_01TM	AG_01TM	...	PS_21TM	AG_21TM	PS_22TM
0	1	1868	지하철	영통	분당선	201906	1165	3061	0	0	...	16245	15177	16339
1	2	4217	지하철	영종	공항철도 1호선	201906	104	2124	0	0	...	1764	6564	1197
2	3	228	지하철	서울대입구(관악구청)	2호선	201906	4793	25645	0	4911	...	58950	100087	55238

Part 2 데이터 설명

공간데이터
<spatial data>

우수 중소 기업
공간데이터

```
# 우수중소기업 공간데이터
EXC = DBF('data/우수중소기업 공간데이터/TL_EXC_SMLPZ_2020.dbf')
EXC = pd.DataFrame(EXC)
```

```
EXC.head(3)
```

	SMLPZ_ID	DEC_SE	CMPNY_NM	RPRSNTV_NM	INDUTY	FOND_DATE	ADDRES	TM_X	TM_Y
0	1	하이브랜드 서울기업	벨금속공업㈜	이희평	날붙이 제조업	1978-01-12	서초구 효령로68길 92 (서초동)	201762.893208	542899.588742
1	2	하이브랜드 서울기업	㈜일성	지영배	액체 펌프 제조업	2000-01-03	구로구 경인로63길 21-6	189818.338312	545438.678543
2	3	하이브랜드 서울기업	이앤에이치㈜	박대전	건설업본사	1998-05-01	성동구 성수이로 7길 27, 701 (서울숲 코오롱디지털타워2차)	204724.207199	549119.988156

Part 2 데이터 설명

공간데이터
<spatial data>

서울시 주요시설, 집객시설
공간데이터

```
# 서울시 주요시설, 집객시설 공간데이터
TBVIATR = DBF('data/주요시설과 집객시설/TL_TBVIATR_FCLTY_INFO_2020.dbf')
TBVIATR = pd.DataFrame(TBVIATR)
TBVIATR.head(3)
```

	STDR_YM	VIAT_SE_CD	VIAT_CD	VIAT_ID	LCLASDC	MLSFCDC	SCLASDC	DCLASDC	BCLASDC	VIAT_NM	ADRES_CD	ADRES_NM
0		030102	0200	AA0000000001	교육/보건	교육시설	초등교육기관	공립초등학교		강덕초교, 고덕그라시움버스정류장쉘터		
1		070301	0302	AA0000000002	산업	서비스산업	종합상품판매업	대형상가	일반종합상가	고덕그라시움상가		
2		030102	0200	AA0000000003	교육/보건	교육시설	초등교육기관	공립초등학교		서울강덕초등학교		

Part 2 데이터 설명

공간데이터
<spatial data>

수도권 지하철역, 공간데이터 (19년)

공간데이터

```
import pandas as pd

# 수도권 지하철역 공간데이터 19년
STATN19 = DBF('./data/수도권지하철 공간데이터/TB_O_SB_STATN_2019.dbf')
STATN19 = pd.DataFrame(STATN19)
STATN19.head(3)
```

	ID	STN_IDN	TGRP_CODE	STN_NM	BB_RT	US_YY_DD	TM_X	TM_Y
0	103.0	416	지하철	미아사거리	4	201911	202653.1120	557083.2120
1	104.0	417	지하철	길음	4	201911	202211.9992	555986.0728
2	105.0	418	지하철	성신여대입구	4	201911	201448.4756	554789.2392

Part 3

데이터 전처리 및 진행 상황

Part 3 데이터 전처리

- 데이터에 포함되어있는 Noise, Outlier, Null을 파악하고 제거해준다.
- '서울시 지하철 30분 단위 이용통계'의 데이터 세트를 확인해 보았다.
- Na값은 없었으며 데이터 분석에 방해되는 백틱을 삭제해 주었다.
- 데이터에 용량이 너무 커 데이터의 형태를 변환하여 데이터의 용량을 줄여주었다.

```
subway20.isna().sum()
```

```
YEAR          0
MONTH         0
DAY           0
HOUR          0
HALF_HOUR     0
LINE_NM       0
STATION_NM    0
GETON_CNT     0
GETOFF_CNT    0
dtype: int64
```

```
# 데이터 양 끝 문자 삭제
subway20 = subway20.apply(lambda x: x.str.strip("`"), axis = 0)
subway21 = subway21.apply(lambda x: x.str.strip("`"), axis = 0)
subway22 = subway22.apply(lambda x: x.str.strip("`"), axis = 0)
```

	YEAR	MONTH	DAY
0	`2020`	`01`	`01`
1	`2020`	`01`	`01`
2	`2020`	`01`	`01`

→

	YEAR	MONTH	DAY
0	2020	01	01
1	2020	01	01
2	2020	01	01

```
subway20['YEAR'] = subway20['YEAR'].astype('int16')
subway20['GETON_CNT'] = subway20['GETON_CNT'].astype('int16')
subway20['GETOFF_CNT'] = subway20['GETOFF_CNT'].astype('int16')
subway20['MONTH'] = subway20['MONTH'].astype('int8')
subway20['DAY'] = subway20['DAY'].astype('int8')
subway20['HOUR'] = subway20['HOUR'].astype('int8')
subway20['HALF_HOUR'] = subway20['HALF_HOUR'].astype('int8')
subway20['STATION_NM'] =  subway20['STATION_NM'].astype('category')
subway20['LINE_NM'] = subway20['LINE_NM'].astype('category')
```

```
subway21['YEAR'] = subway21['YEAR'].astype('int16')
subway21['GETON_CNT'] = subway21['GETON_CNT'].astype('int16')
subway21['GETOFF_CNT'] = subway21['GETOFF_CNT'].astype('int16')
subway21['MONTH'] = subway21['MONTH'].astype('int8')
subway21['DAY'] = subway21['DAY'].astype('int8')
subway21['HOUR'] = subway21['HOUR'].astype('int8')
subway21['HALF_HOUR'] = subway21['HALF_HOUR'].astype('int8')
subway21['STATION_NM'] =  subway21['STATION_NM'].astype('category')
subway21['LINE_NM'] = subway21['LINE_NM'].astype('category')
```

```
subway22['YEAR'] = subway22['YEAR'].astype('int16')
subway22['GETON_CNT'] = subway22['GETON_CNT'].astype('int16')
subway22['GETOFF_CNT'] = subway22['GETOFF_CNT'].astype('int16')
subway22['MONTH'] = subway22['MONTH'].astype('int8')
subway22['DAY'] = subway22['DAY'].astype('int8')
subway22['HOUR'] = subway22['HOUR'].astype('int8')
subway22['HALF_HOUR'] = subway22['HALF_HOUR'].astype('int8')
subway22['STATION_NM'] =  subway22['STATION_NM'].astype('category')
subway22['LINE_NM'] = subway22['LINE_NM'].astype('category')
```

Part 3

데이터 전처리

30분단위 이용 통계 → csv 파일

```
memory usage: 323.6+ MB
```

```
memory usage: 58.4 MB
```

- 용량이 크기에 데이터 형식을 바꾸어 데이터의 용량을 줄임
- 월별로 나눠져 있는 데이터를 년도별로 합침, 데이터에 필요 없는 문자(`) 삭제

서울시 지하철 시간대별 승객 수 → dbf 파일

우수중소기업 공간데이터 → dbf 파일

수도권지하철 공간데이터 → dbf 파일

서울시 주요시설과 집객시설 공간데이터 → dbf 파일

Part 3 진행 상황

```
# 데이터셋 불러오기

# subway데이터셋: 서울시 30분 단위 이용 통계(20~22년)
subway20 = pd.read_csv("data/subway20_df.csv")
subway21 = pd.read_csv("data/subway21_df.csv")
subway22 = pd.read_csv("data/subway22_df.csv")

# EXC데이터셋: 우수 중소기업 공간데이터 20년도
EXC =  pd.read_csv("data/EXC_df.csv")

# TBVIATR데이터셋: 서울시 주요시설, 집객시설 주요데이터(20년)
TBVIATR = pd.read_csv("data/TBVIATR_df.csv")

# Seoul데이터셋: 서울시 지하철 시간대별 승객수(16~19년)
SEOUL17 = pd.read_csv("data/SEOUL17_df.csv")
SEOUL18 = pd.read_csv("data/SEOUL18_df.csv")
SEOUL19 = pd.read_csv("data/SEOUL19_df.csv")

# STATN데이터셋: 수도권 지하철 공간데이터(19년)
STATN19 = pd.read_csv("data/STATN19_df.csv")
```

데이터 셋 불러오기

- subway 데이터셋: 서울시 30분 단위 이용 통계(20~22년)
- EXC 데이터셋: 우수 중소기업 공간데이터 20년도
- TBVIATR 데이터셋: 서울시 주요시설, 집객시설 주요데이터(20년)
- Seoul 데이터셋: 서울시 지하철 시간대별 승객수(16~19년)
- STATN 데이터셋: 수도권 지하철 공간데이터(19년)

Part 3 진행 상황

1. 시각화

```
# 파이썬 시각화 패키지 불러오기
import matplotlib.pyplot as plt
%matplotlib inline


# 사용자 운영체제 확인
import platform
platform.system()

# 운영체제별 한글 폰트 설정
if platform.system() == 'Darwin': # Mac 환경 폰트 설정
    plt.rc('font', family='AppleGothic')
elif platform.system() == 'Windows': # Windows 환경 폰트 설정
    plt.rc('font', family='Malgun Gothic')

plt.rc('axes', unicode_minus=False) # 마이너스 폰트 설정


# 글씨 선명하게 출력하는 설정
%config InlineBackend.figure_format = 'retina'
```

Part 3 진행 상황

호선별 승하차 인원 합계 시각화

호선별 승하차인원 합계 시각화

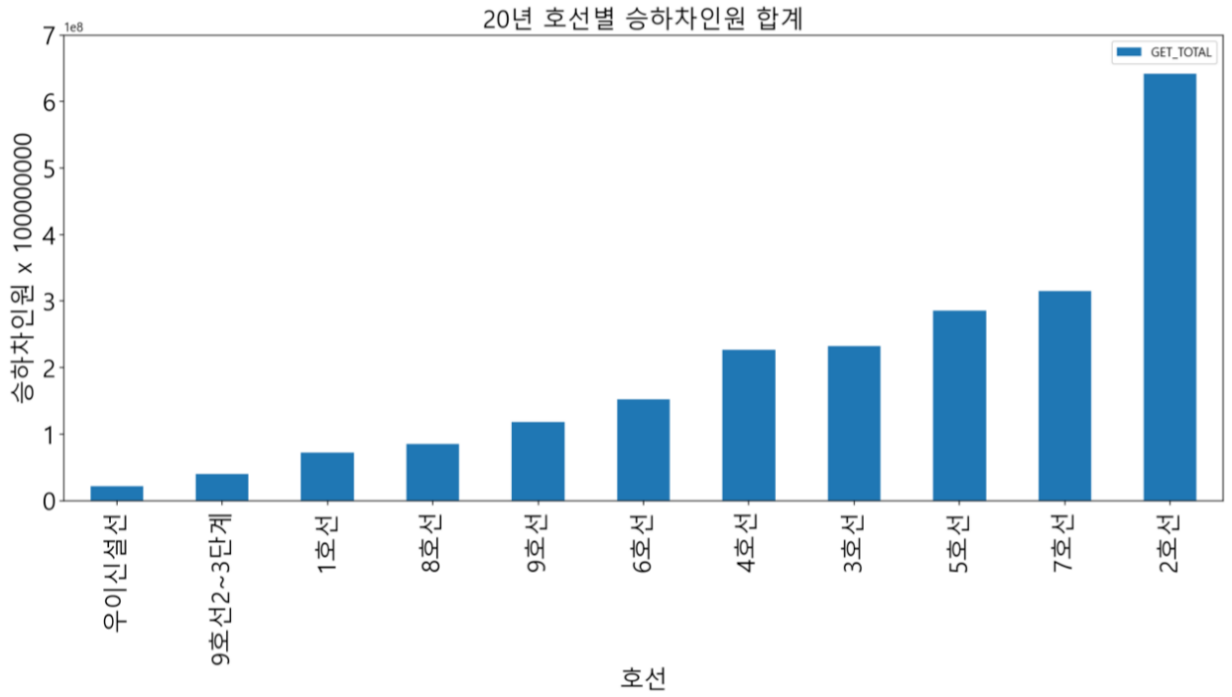
```
subway20['GET_TOTAL'] = subway20['GETON_CNT'] + subway20['GETOFF_CNT']
```

```
subway20_total = subway20[['LINE_NM','GET_TOTAL']].groupby('LINE_NM', as
subway20_total
```

	LINE_NM	GET_TOTAL
10	우이신설선	21776882
9	9호선2~3단계	40225256
0	1호선	72162962
7	8호선	84974557
8	9호선	117855526
5	6호선	152601931
3	4호선	227004039
2	3호선	232190958
4	5호선	286124371
6	7호선	315170457
1	2호선	641661884

```
import matplotlib.pyplot as plt
# fig, ax = plt.subplots(figsize = (12, 12))
subway20_total.plot.bar(x = 'LINE_NM', y = 'GET_TOTAL', fontsize = 20, figsize = (17, 7))
plt.xlabel('호선', fontsize = 20)
plt.ylabel('승하차인원 x 100000000', fontsize = 20)
plt.ylim([0, 700000000])
plt.title("20년 호선별 승하차인원 합계", fontsize = 20)
```

```
Text(0.5, 1.0, '20년 호선별 승하차인원 합계')
```



Part 3 진행 상황

역별 승하차 인원 시각화

```
# 역별 승하차 인원 확인
subway20['GET_TOTAL'] = subway20['GETON_CNT'] + subway20['GETOFF_CNT']
subway20
subway20_sort = subway20.sort_values(ascending = False, by = 'GET_TOTAL')
subway20_sort.groupby(['STATION_NM', 'LINE_NM']).sum('GET_TOTAL')[['GET_TOTAL']].
```

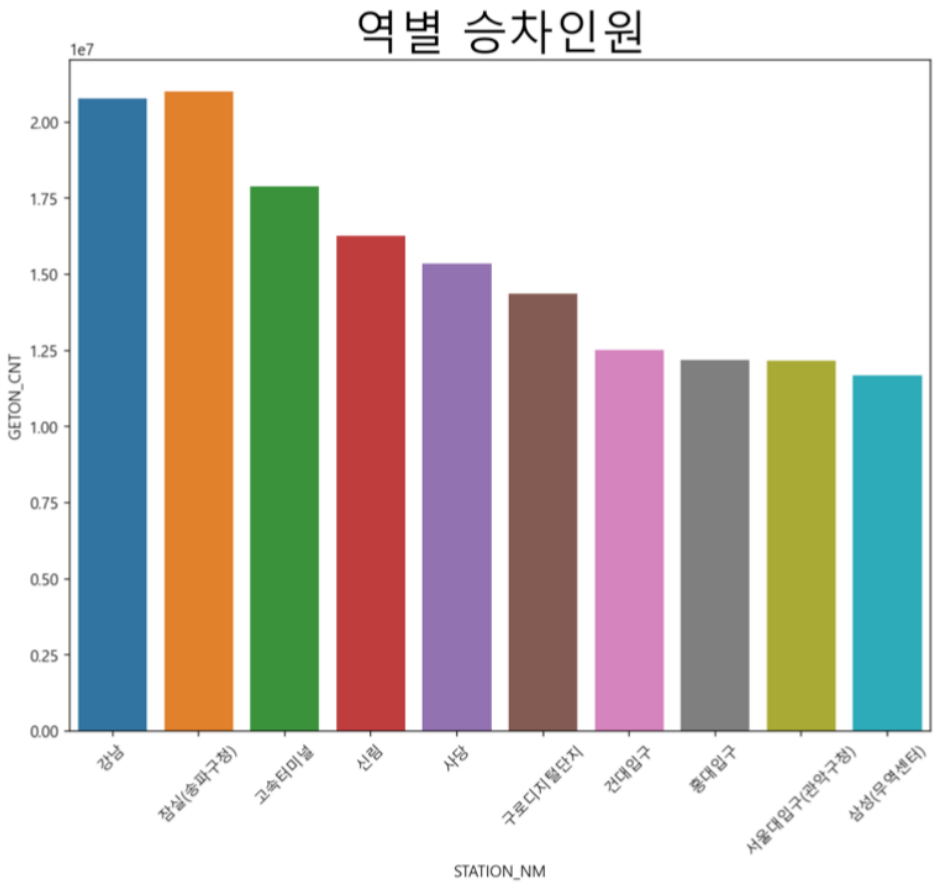
		GET_TOTAL
STATION_NM	LINE_NM	
강남	2호선	42188448
잠실(송파구청)	2호선	34778138
신림	2호선	31905562
구로디지털단지	2호선	28556403
홍대입구	2호선	25287511
서울대입구(관악구청)	2호선	23965740
삼성(무역센터)	2호선	23464178
선릉	2호선	22906969
역삼	2호선	22070781
고속터미널	3호선	21857951
신도림	2호선	21384510
사당	2호선	19717826
건대입구	2호선	19590432
양재(서초구청)	3호선	18941619
을지로입구	2호선	18698688

```
subway20 = subway20.pivot_table(index='STATION_NM', values=['GETON_CNT','GETOFF_CNT'], aggfunc='sum').reset_index()
subway20
subway20_sort = subway20.sort_values(by='GETOFF_CNT', ascending=False)
subway20_sort
```

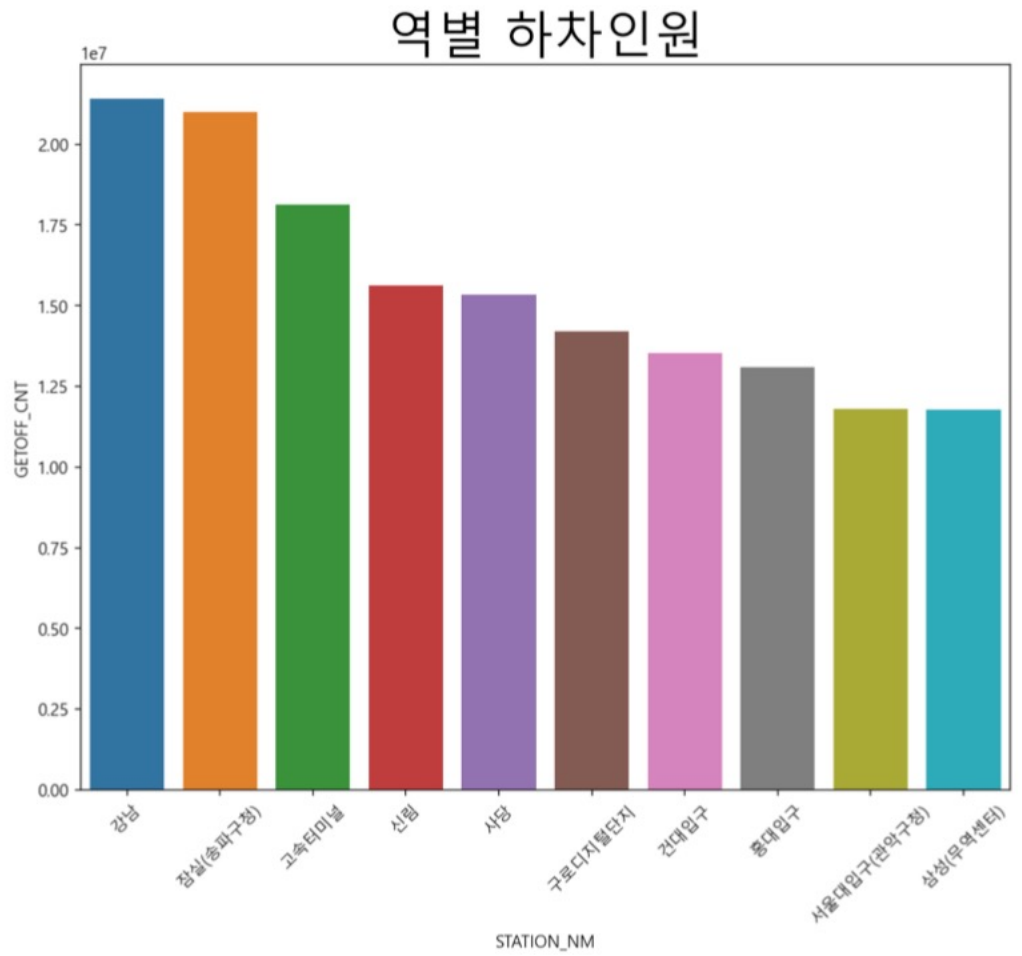
	STATION_NM	GETOFF_CNT	GETON_CNT
5	강남	21400896	20787552
236	잠실(송파구청)	21004378	20994812
19	고속터미널	18144644	17876809
178	신림	15632652	16272910
125	사당	15348165	15349740
...	...	...	...
222	용마산	258554	273103
241	장암	251592	626055
78	둔촌오륜	229237	249536
48	남태령	215328	256613
171	신내	0	28

Part 3 진행 상황

```
import seaborn as sns

sw_s = plt.figure(figsize=(10,8))
plt.xticks(rotation=45)
sw_s = plt.title('역별 승차인원', fontsize=30)
sw_s = sns.barplot(data=subway20_sort.head(10), x='STATION_NM', y='GETON_CNT')

# 강남과 잠실의 순서가 승차인원과 하차인원에서 다르다.
```




```
import seaborn as sns

sw_s = plt.figure(figsize=(10,8))
plt.xticks(rotation=45)
sw_s = plt.title('역별 하차인원', fontsize=30)
sw_s = sns.barplot(data=subway20_sort.head(10), x='STATION_NM', y='GETOFF_CNT')
```



Part 3 데이터 전처리

서울시 30분 이용통계 (20년)

1. 라이브러리 및 데이터 불러오기

```
import pandas as pd
import numpy as np
import glob
import os
```

```
# 2020년 1월부터 12월까지의 csv파일을 하나의 파일로 합치는 작업

input_file = r'data' # csv파일들이 있는 디렉토리 위치
output_file = r'data\subway20.csv' # 병합하고 저장하려는 파일명

allFile_list = glob.glob(os.path.join(input_file, 'TBDM_TRNSIT_STAT_SUBWAY_2020
print(allFile_list)
allData = [] # 읽어 들인 csv파일 내용을 저장할 빈 리스트를 하나 만든다
for file in allFile_list:
    df = pd.read_csv(file) # for구문으로 csv파일들을 읽어 들인다
    allData.append(df) # 빈 리스트에 읽어 들인 내용을 추가한다

dataCombine = pd.concat(allData, axis=0, ignore_index=True) # concat함수를 이용해서
# axis=0은 수직으로 병합함. axis=1은 수평. ignore_index=True는 인데스 값이 기존 순서를 무시하
dataCombine.to_csv(output_file, index=False) # to_csv함수로 저장한다. 인데스를 빼려면 F
```

```
['data\\TBDM_TRNSIT_STAT_SUBWAY_202001.csv', 'data\\TBDM_TRNSIT_STAT_SUBWAY_202
202004.csv', 'data\\TBDM_TRNSIT_STAT_SUBWAY_202005.csv', 'data\\TBDM_TRNSIT_STA
STAT_SUBWAY_202008.csv', 'data\\TBDM_TRNSIT_STAT_SUBWAY_202009.csv', 'data\\TBD
TBDM_TRNSIT_STAT_SUBWAY_202012.csv']
```

```
subway20 = pd.read_csv("data/subway20.csv")
```

데이터 셋 불러오기

- 2020 1월부터 12월까지 csv 파일 합치기
- 2021, 2022년도 같은 방식으로 진행

Part 3

데이터 전처리

2. 데이터 전처리

```
# 필요없다고 생각되는 컬럼 삭제
subway20 = subway20.drop(["`STATION_ID`"], axis = 1)
```

```
# 컬럼에 쓰잘데기없는 기호 `(그 키보드에 물결)이 들어가있어서
# rename 진행

subway20 = subway20.rename({'`YEAR`':'YEAR',
                  '`MONTH`':'MONTH',
                  '`DAY`':'DAY',
                  '`HOUR`':'HOUR',
                  '`HALF_HOUR`':'HALF_HOUR',
                  '`LINE_NM`':'LINE_NM',
                  '`STATION_NM`':'STATION_NM',
                  '`GETON_CNT`':'GETON_CNT',
                  '`GETOFF_CNT`':'GETOFF_CNT'}, axis='columns')
subway20.head(3)
```

	YEAR	MONTH	DAY	HOUR	HALF_HOUR	LINE_NM	STATION_NM	GETON_CNT	GETOFF_CNT
0	`2020`	`01`	`01`	`04`	`30`	`1호선`	`서울역`	`11`	`1`
1	`2020`	`01`	`01`	`04`	`30`	`1호선`	`시청`	`1`	`0`
2	`2020`	`01`	`01`	`04`	`30`	`1호선`	`종각`	`5`	`2`

```
# value값에 쓰잘데기없는 `기호 없애기

subway20 = subway20.apply(lambda x: x.str.strip("`"), axis = 0)
subway20.head(3)
```

	YEAR	MONTH	DAY	HOUR	HALF_HOUR	LINE_NM	STATION_NM	GETON_CNT	GETOFF_CNT
0	2020	01	01	04	30	1호선	서울역	11	1
1	2020	01	01	04	30	1호선	시청	1	0
2	2020	01	01	04	30	1호선	종각	5	2

컬럼 삭제 및 기호 삭제

- STATION_ID(역 아이디) 제거
- Value 값에 쓰잘데기 없는 ` 기호 없애기

Part 3

데이터 전처리

```
# 데이터셋 정보 확인
subway20.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4712377 entries, 0 to 4712376
Data columns (total 9 columns):
 #   Column      Dtype
---  ------      -----
 0   YEAR        object
 1   MONTH       object
 2   DAY         object
 3   HOUR        object
 4   HALF_HOUR   object
 5   LINE_NM     object
 6   STATION_NM  object
 7   GETON_CNT   object
 8   GETOFF_CNT  object
dtypes: object(9)
memory usage: 323.6+ MB
```

```
# 형 변환전 기술통계량 확인

subway20.describe().T
```

	count	unique	top	freq
YEAR	4712377	1	2020	4712377
MONTH	4712377	12	01	407567
DAY	4712377	31	27	155342
HOUR	4712377	24	18	239251
HALF_HOUR	4712377	2	00	2372583
LINE_NM	4712377	11	5호선	744529
STATION_NM	4712377	289	신설동	43775
GETON_CNT	4712377	5960	0	76975
GETOFF_CNT	4712377	6378	0	124402

```
print(np.iinfo('int16'))
print(np.iinfo('int8'))
```

```
Machine parameters for int16
---------------------------------------------------------------
min = -32768
max = 32767
---------------------------------------------------------------

Machine parameters for int8
---------------------------------------------------------------
min = -128
max = 127
---------------------------------------------------------------
```

Part 3

데이터 전처리

```
subway20['YEAR'] = subway20['YEAR'].astype('int16')  # 2020이므로 int8 불가
subway20['MONTH'] = subway20['MONTH'].astype('int8')
subway20['DAY'] = subway20['DAY'].astype('int8')
subway20['HOUR'] = subway20['HOUR'].astype('int8')
subway20['HALF_HOUR'] = subway20['HALF_HOUR'].astype('int8')
subway20['GETON_CNT'] = subway20['GETON_CNT'].astype('int16') #int8 불가
subway20['GETOFF_CNT'] = subway20['GETOFF_CNT'].astype('int16') #int8 불가
subway20['LINE_NM'] = subway20['LINE_NM'].astype('category')
subway20['STATION_NM'] =  subway20['STATION_NM'].astype('category')
```

```
# 최종 용량 확인
subway20.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4712377 entries, 0 to 4712376
Data columns (total 9 columns):
 #   Column      Dtype
---  ------      -----
 0   YEAR        int16
 1   MONTH       int8
 2   DAY         int8
 3   HOUR        int8
 4   HALF_HOUR   int8
 5   LINE_NM     category
 6   STATION_NM  category
 7   GETON_CNT   int16
 8   GETOFF_CNT  int16
dtypes: category(2), int16(3), int8(4)
memory usage: 58.4 MB
```

```
subway20.to_csv("subway20_df.csv", index = False)
```

데이터 형식 변경

- Int 와 category 로 변경
- 최종 용량 확인

우수 중소 기업 공간데이터

공간데이터(20년)

```
!pip install dbfread
```

```
Requirement already satisfied: dbfread in c:\users\jemin\anaconda3\lib\site-packages (2.0.7)
```

```
from dbfread import DBF
```

```
# 우수중소기업 공간데이터
EXC = DBF('data/TL_EXC_SMLPZ_2020.dbf')
EXC = pd.DataFrame(EXC)
```

```
EXC.head(3)
```

	SMLPZ_ID	DEC_SE	CMPNY_NM	RPRSNTV_NM	INDUTY	FOND_DATE	ADDRES	TM_X	TM_Y
0	1	하이브랜드서울기업	벨금속공업㈜	이희평	날붙이 제조업	1978-01-12	서초구 효령로68길 92 (서초동)	201762.893208	542899.588742
1	2	하이브랜드서울기업	㈜일성	지영배	액체 펌프 제조업	2000-01-03	구로구 경인로63길 21-6	189818.338312	545438.678543
2	3	하이브랜드서울기업	이앤에이치㈜	박대전	건설업본사	1998-05-01	성동구 성수이로 7길 27, 701 (서울숲 코오롱디지털타워2차)	204724.207199	549119.988156

```
EXC.to_csv("EXC_df.csv", index=False)
```

Part 3

데이터 전처리

서울시 주요시설, 집객시설 공간데이터(20년)

```
# 서울시 주요시설, 집객시설 공간데이터
TBVIATR = DBF('data/TL_TBVIATR_FCLTY_INFO_2020.dbf')
TBVIATR = pd.DataFrame(TBVIATR)
TBVIATR.head(3)
```

	STDR_YM	VIAT_SE_CD	VIAT_CD	VIAT_ID	LCLASDC	MLSFCDC	SCLASDC	DCLASDC	BCLASDC	VIAT_NM	AD
0		030102	0200	AA0000000001	교육/보건	교육시설	초등교육기관	공립초등학교		강덕초교, 고덕그라시움버스정류장쉘터	
1		070301	0302	AA0000000002	산업	서비스산업	종합상품판매업	대형상가	일반종합상가	고덕그라시움상가	
2		030102	0200	AA0000000003	교육/보건	교육시설	초등교육기관	공립초등학교		서울강덕초등학교	

```
SEOUL17.to_csv("SEOUL17_df.csv", index = False)
SEOUL18.to_csv("SEOUL18_df.csv", index = False)
SEOUL19.to_csv("SEOUL19_df.csv", index = False)
```

서울시 지하철 시간대별 승객수(16~19년)

```
# 서울시 지하철 시간대별 승객수 16년
SEOUL16 = DBF('data/TB_SEOUL_TRAIN_PASSN_2016.dbf')
SEOUL16 = pd.DataFrame(SEOUL16)
SEOUL16.head(3)
```

	ID	STN_IDN	TGRP_CODE	STN_NM	BB_RT	US_YY_DD	PS_00TM	AG_00TM	PS_01TM	AG_01TM	...	AG_20TM	PS_21TM	AG_21TM	P
0	0	415	지하철	미아	4	201511	650	5420	0	37	...	36798	15898	32859	
1	0	416	지하철	미아사거리	4	201511	2079	8760	0	0	...	75753	29959	65909	
2	0	417	지하철	길음	4	201511	1466	8951	0	0	...	59255	23816	54402	

3 rows × 57 columns

Part 3

데이터 전처리

수도권 지하철 공간데이터(19년)

```
import pandas as pd

# 수도권 지하철역 공간데이터 19년
STATN19 = DBF('./data/TB_O_SB_STATN_2019.dbf')
STATN19 = pd.DataFrame(STATN19)
STATN19.head(3)
```

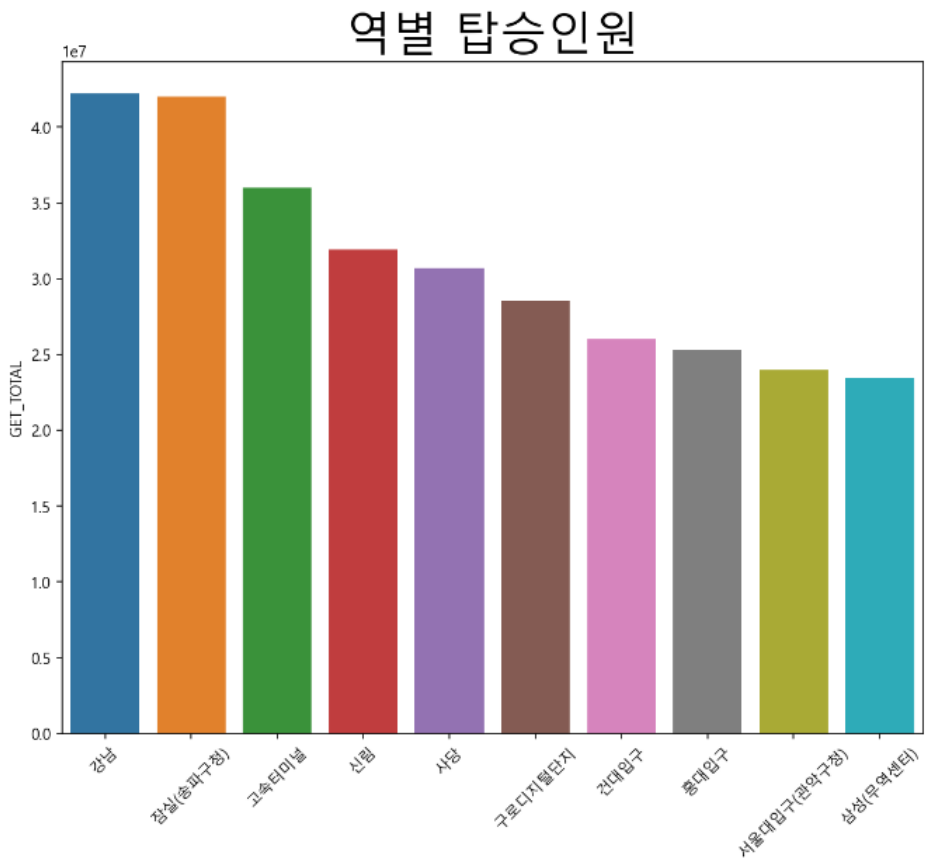
	ID	STN_IDN	TGRP_CODE	STN_NM	BB_RT	US_YY_DD	TM_X	TM_Y
0	103.0	416	지하철	미아사거리	4	201911	202653.1120	557083.2120
1	104.0	417	지하철	길음	4	201911	202211.9992	555986.0728
2	105.0	418	지하철	성신여대입구	4	201911	201448.4756	554789.2392

```
STATN19.to_csv("STATN19_df.csv", index = False)
```

Part 3

데이터 전처리

```
import seaborn as sns

sw_s = plt.figure(figsize=(10,8))
plt.xticks(rotation=45)
sw_s = plt.title('역별 탑승인원', fontsize=30)
sw_s = sns.barplot(data=subway20_sort.head(10), x='STATION_NM', y='GET_TOTAL')

# 강남과 잠실의 순서가 승차인원과 하차인원에서 다르다.
```



2020 역별 탑승 인원

- 하차인원 + 승차인원

Part 4

서비스 방안

Part 4

서비스 방안

1. **버스 추가 배차는 급행버스를 대상으로 고터, 신림, 사당, 구로디지털단지 순으로..**

많은 버스 노선들 중 급행을 선택한 이유는 승객들의 정차 없는 빠른 출퇴근을 위해서입니다. 긴 거리를 이동하는데 들리는 정류장이 많다면 정차 후 승객들을 태우고 내리는 시간이 소모 될 것입니다.

그렇기에 급행 열차 처럼 지하철 이용객들이 가장 많은 잠실, 강남에서 사람들을 태워 고터, 신림, 사당, 구로디지털단지와 같이 승하차 인원이 집중되는 역에서 하차를 해주는 방법으로 추가 급행버스 배차를 해줍니다.

2. 지하철로 몰리는 인구를 왜 분산하려고 하는가?

한정된 인원과 자원을 최대한 활용을 하여 최상의 결과를 내는 것이 이번 데이터 분석의 목표였습니다. 사용인원이 많은 노선, 역을 파악하고 하루 중 인원이 집중되는 시간대, 일 주일 중 인원이 집중되는 요일에 추가배차를 하거나 버스와 같은 기타 교통수단으로의 사용 인원 분담을 유도한다면 필요 최소한의 투입으로 효율성을 증진시키는 것에 이바지 할 것입니다.

3. 지하철로 몰리는 인구를 왜 분산하려고 하는가?

[어플개발??]

시간대별 혼잡도를 분석해서 현재 만약 해당 지하철에 사람이 매우 붐빈다면, 어플에서 알림을 준다. 예를들어 행사에 참여를 하시겠습니까?(O / X)에서 O를 눌러 행사에 참여한다면 서울시에서 참여한 인원들에게 포인트를 준다. 근데 이제 이 포인트를 나중에 지역화폐로 환급해준다.(혹은 다른 문화거리 사용가능한 코인?) 가장 중요한것은 지하철 탑승대신 행사에 참여한 사람들이 있기 때문에 그만큼 해당 시간에 지하철 이용객이 조금 더 줄 것이다.

3. 지하철로 몰리는 인구를 왜 분산하려고 하는가?

1-1~1-4과 같이 특정 번호칸을 예약제 시스템 도입(혼잡한 시간에만)

4. 지하철 혼잡도 분석에 따른 건의사항 (지하철 노선, 지하철 간격)

평소 일관성 있는 혼잡도 외에 추가로 특정 날에 많은 이동이 있다면 이것이 결론에 영향을 안미친다 할 수 있을까요?

위에 처럼 자세한 분석이 가능하다면 이를 통해 특정 날(분석결과 6/2일만 이상하게 특정역에 사람이 쏠린다?) 지하철 시간을 유동적으로 변경하는 방향으로 건의 혹은 이를 통해 정확한 결론으로 뭐를

해야한다 이런것도 좋지만, 위에처럼 데이터 분석해서 어느역이, 언제, 몇시 쯤 혼잡한지 분석을 통해서 정확히 알아내는 것을 목표로 하는 것은 어떠할까요?

추가로 분석 진행중 인사이트를 얻으면 그것에 대해서 더 생각해 봐야할 것 같습니다.

Part 5

질문

Question

- 앞서 설명한 서비스 방안에 대한 피드백을 받고 싶습니다.

- 분석을 통해 인사이트를 얻는 것 말고 다양한 학습(시계열 분석 및 머신러닝)을 시키는 방향으로도 진행가능한 부분인지 궁금합니다.

- 저희는 2호선 데이터(가 역대급으로 많아서)만 가지고 진행할 예정입니다.

  어떤가요? 의견 부탁드립니다.

Question

- 공간데이터 좌표 변환이 안된다.
  - 공간데이터의 좌표가 TM중부원점으로 되어있다.
  - 변환을 시키는 과정에서 잘못된 위도,경도가 도출된다.

- 더 큰 데이터를 사용해야 할지?
  - 30분 단위 데이터 말고 다른 데이터가 있긴 한데 이 데이터가 추가로 더 필요할까요 ?

Part 6

향후 계획

Part 6

향후 계획

- 어느 역, 특정 요일, 혹은 특정 시간을 봐주었다. 이에 따른 가장 혼잡한 역을 선정, 주변 시설과의 연관성 확인한다.
- 혼잡도가 주변 시설과의 상관이 있다면 그 이유를 파악한다.
- 가능하다면 특별한 날, 시간 등에 따른 혼잡도 까지 세세한 분석을 통해 확인 한다.

  - 데이터가 추가로 필요할지?

Thank you

극데노 (극한의 데이터 노예)
유제우
이서영
이현중