

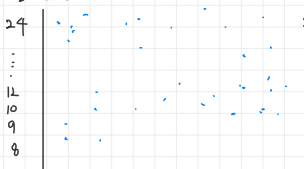
30분 단위 data → 시각화

· 시간별 혼잡 상황 (줄알지, 도카지)

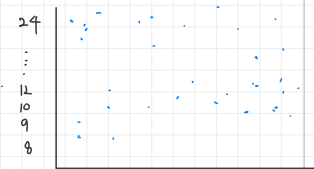
년도 / 월 / 일 / 시간 / 30분 / 호선 / 역 / 승차인원
/ 하차인원

* 호선 별 역

<1호선>



<2호선>



서울역 종로3가 시청 전대입구 ... 성사 사당 강남역 신도림 ...

* 시각들이 많이 이용하는 지하철

■ 2018년 일평균 승하차인원 순위(서울교통공사 수상실적)

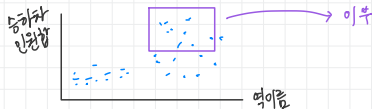
순위	역명	일평균 승하차인원	순위	역명	일평균 승하차인원
1	강남	204,144	11	신동	101,233
2	광명2	175,376	12	물지로입구	98,762
3	홍대입구	165,072	13	신촌	98,409
4	신림	138,692	14	역삼	96,490
5	구로디지털단지	124,380	15	전대입구2	93,505
6	고속터미널3	120,887	16	강변	92,784
7	상정	119,572	17	사당2	91,263
8	신도림	119,414	18	해라	85,496
9	서울역(1)	106,475	19	종각	85,370
10	서울대입구	105,330	20	연신내3	82,994

서울시 지하철 30분 단위 이용 통계

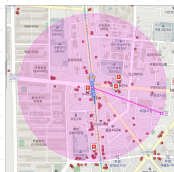
→ 승하차 인원 SUM 값 통계 내서

subway_use_best = subway_use.sort_values(ascending = False, by = '승하차인원 합').head(10)

plt.figure(figsize=[10,10])
sns.scatterplot(data=subway_use_best, x='역이름', y='승하차인원 합')



<공간 Data>



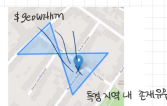
역 주변 주요시설, 중소기업 공간데이터

이용해서 2dsphere 써서

← 이론적으로 나타내면 좋을 듯

2dsphere 관련 연산자들

- \$near: 거리 계산해 가까운 곳에서 먼 곳의 결과 도출
- \$geoNear: near 와 유사하나, aggregation operator (집계연산자)
- \$nearSphere: 특정 거리 내에 존재하는지 확인
- \$geoWithin: 특정 지역 안에 존재하는지 확인
- \$geoIntersects: 특정 지역을 포함하는지 확인
- > 통과하다



효율적인 지하철 인원 분배

- 데이터 전처리
- 지하철 역별 승·하차 인원 시각화
- 시간대별 노선 승·하차 인원 추이 시각화

2020-03-01 → 묶음 건지

그룹화를 할건지

```
data_station = raw.pivot_table(index = ['지하철 역'], values = '하차인원',
aggfunc='sum').astype('int')
```

```
data_station1 = raw.pivot_table(index = ['지하철 역'], values = '승차인원', aggfunc='sum').astype('int')
```

```
a = data_station.sort_values(by="하차인원",ascending=False).head(11).reset_index()
```

```
aa=data_station1.sort_values(by="승차인원",ascending=False).head(11).reset_index()
```

1000 만명 이상인 역 대상 → 선정해서 조사

```
df_sub_station = a[a['하차인원']>=10000000]
```

```
df_sub_station2 = aa[aa['승차인원']>=10000000]
```

```
df_sub_station2
```

지하철 역	승·하차 인원
잠실	.
강남	.
고덕역마차	.
...	.

분석 대상 되는 지하철 역 list 만들기

```
list_df=df_sub_station['지하철 역'].to_list()
list_df[0]
```

```
sub_df = pd.DataFrame()
```

```
for i in list_df:
```

```
    a =raw[raw['지하철 역']==i]
```

```
    sub_df = sub_df.append(a)
```

```
sub_df
```

일자	노선	역	승차인원	하차인원	요일
----	----	---	------	------	----

* 시각화

서울시 역별 하차인원, 승차인원

```
cd= sub_df.pivot_table(index='지하철 역',values=['승차인원','하차인원'],aggfunc='sum').reset_index()
cd
```

```
cd =cd.sort_values(by='하차인원',ascending=False)
```

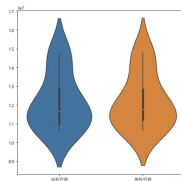
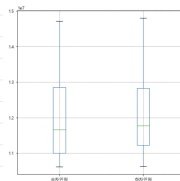
```
plt.figure(figsize=(15,15))
```

```
plt.subplot(221)
```

```
cd.boxplot()
```

```
plt.subplot(222)
```

```
sns.violinplot(data=cd)
```



역별

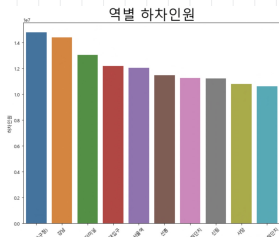
```
acd=cd.sort_values(by='승차인원',ascending=False).reset_index()
acd
```

```
bunmo=acd['승차인원'].sum()
```

```
ratio1=acd['승차인원']/bunmo
ratio1
```

```
ratio2=acd['하차인원']/bunmo
ratio2
```

```
_=plt.figure(figsize=(10,8))
plt.xticks(rotation=45)
plt.title('역별 하차인원',fontsize=30)
_=sns.barplot(data=cd.head(10), x='지하철 역', y='하차인원')
```

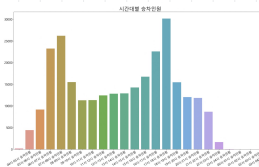


시간대별

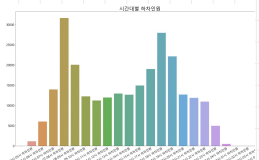
```
c=timeraw.filter(regex='하차')
```

```
d=timeraw.filter(regex='승차')
```

```
#시간대별 승차 인원
plt.figure(figsize=(27,15))
a=sns.barplot(data=d,ci=False)
_=plt.xticks(rotation=30,fontsize=18)
_=plt.yticks(fontsize=18)
_=plt.title('시간대별 승차인원',fontsize=30)
```



```
#시간대별 하차 인원
plt.figure(figsize=(27,15))
a=sns.barplot(data=c,ci=False)
_=plt.xticks(rotation=30,fontsize=18)
_=plt.yticks(fontsize=18)
_=plt.title('시간대별 하차인원',fontsize=30)
```

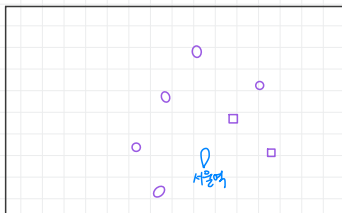


지도로 시각화

→ 공간데이터 위도, 경도 사용하면 되려나?

질 문캠용 T.O.T

- 서울시 지하철 역 공간 data
- 서울시 주요시설 공간 data



```
import folium
from folium.plugins import MiniMap
from folium.plugins import MarkerCluster
```

```
bs= folium.Map(location = ['37.5536067','126.9674308'], zoom_start = 15)
```

```
bs= folium.Map(location = ['37.5536067','126.9674308'], zoom_start = 15)
marker_cluster = MarkerCluster().add_to(bs)
for i in range(len(bus_site_df)):
    long = bus_site_df.loc[i,'X좌표']    #loc['행 번호','열 이름']
    lat = bus_site_df.loc[i,'Y좌표']
    name = bus_site_df.loc[i,'정류소명']
    name1=bus_site_df.loc[i,'정류소번호']
    folium.Marker([lat, long],tooltip=name,popup=
name1).add_to(marker_cluster)
bs
```

이렇게 역 주변 몇개해서 → 지도 합치기



```
bs=folium.Map(location = ['37.5536067','126.9674308'], zoom_start = 15)
marker_cluster = MarkerCluster().add_to(bs)
mymap = MiniMap()
mymap.add_to(bs)
for i in range(len(bus_site_df)):
    long = bus_site_df.loc[i,'X좌표']    #loc['행 번호','열 이름']
    lat = bus_site_df.loc[i,'Y좌표']
    name = bus_site_df.loc[i,'정류소명']
    name1=bus_site_df.loc[i,'정류소번호']
    folium.Marker([lat, long],tooltip=name,popup=
name1).add_to(marker_cluster)
bs

for i in range(len(bus_site_df)):
    lat = bus_site_df.loc[i,'Y좌표']
    long = bus_site_df.loc[i,'X좌표']
    name = bus_site_df.loc[i,'정류소명']
    name1=bus_site_df.loc[i,'정류소번호']
    folium.Marker([lat, long],tooltip=name+'X좌표'+name1).add_to(bs)
    folium.CircleMarker([lat, long],radius=100,color='yellow').add_to(bs)
```