

# Python 프로젝트 진행 보고서

프로젝트 이름	확진자 동선 분석을 통한 방역 강화 업종 선별	팀 이름	3팀
팀원	우승엽(팀장), 박나은, 이광현, 박소연	수행기간	2021.05.18(화) ~ 2021.05.24(월)
진행 내용	<ul style="list-style-type: none"><li>수집한 확진자 이동동선 데이터를 바탕으로 장소유형(업소) 분류 작업</li><li>분류한 데이터를 가독성 좋은 시각적 통계로 구현</li></ul>		
차주 수행 계획	<ul style="list-style-type: none"><li>각 팀원이 조사한 할당지역의 확진자 이동경로 엑셀파일을 통일성 있게 수정하여, 각각의 엑셀파일을 하나로 병합하는 작업 수행 (현재는 각자 수집한 데이터를 엑셀파일에서 수동으로 복사하여 하나의 엑셀 파일에 합침)</li><li>최종 실행에서 오류가 없는지 점검하고 오류가 있다면 수정하여 완성시킨다.</li></ul>		

## <별첨>

수행 내용에 대한 부연 설명이 필요한 경우 작성합니다. (스크린 샷 등)

```
elif "마트" in sheet["B" + str(i)].value or "판매점" in sheet["B" + str(i)].value:
    store += 1
    sum += 1
elif "오락문화" in sheet["B" + str(i)].value or "노래방" in sheet["B" + str(i)].value:
    songroom += 1
    sum += 1
elif "직장" in sheet["B" + str(i)].value or "사무시설" in sheet["B" + str(i)].value:
    work += 1
    sum += 1
elif "학교" in sheet["B" + str(i)].value or "어린이집" in sheet["B" + str(i)].value:
    school += 1
    sum += 1
elif "학원" in sheet["B" + str(i)].value:
    academy += 1
    sum += 1
else: # 장소유형 미공개 데이터 기타로 처리
    etc += 1

loc = [{"PC방", pcroom},
       ["카페", cafe],
       ["놀이공원", playpark],
       ["교통수단", transportation],
       ["음식점", restaurant],
       ["주민센터", com_center],
       ["병원", hospital],
       ["공중위생업(목욕탕, 미용실 등)", publicSanitation],
```

- 수집한 확진자 이동동선 데이터를 장소유형(업소)별로 분류

```
from test import loc
from operator import itemgetter
loc.sort(key=itemgetter(1)) # 장소유형 횡

frequency = [] #빈도수
labels = [] #장소유형

for i in range(0, len(loc)):
    frequency.append(loc[i][1])
    labels.append(loc[i][0])

## 데이터 라벨, 빈도수, 색상을 빈도수를 가
labels_frequency = zip(labels, frequency)
labels_frequency = sorted(labels_frequency)
```

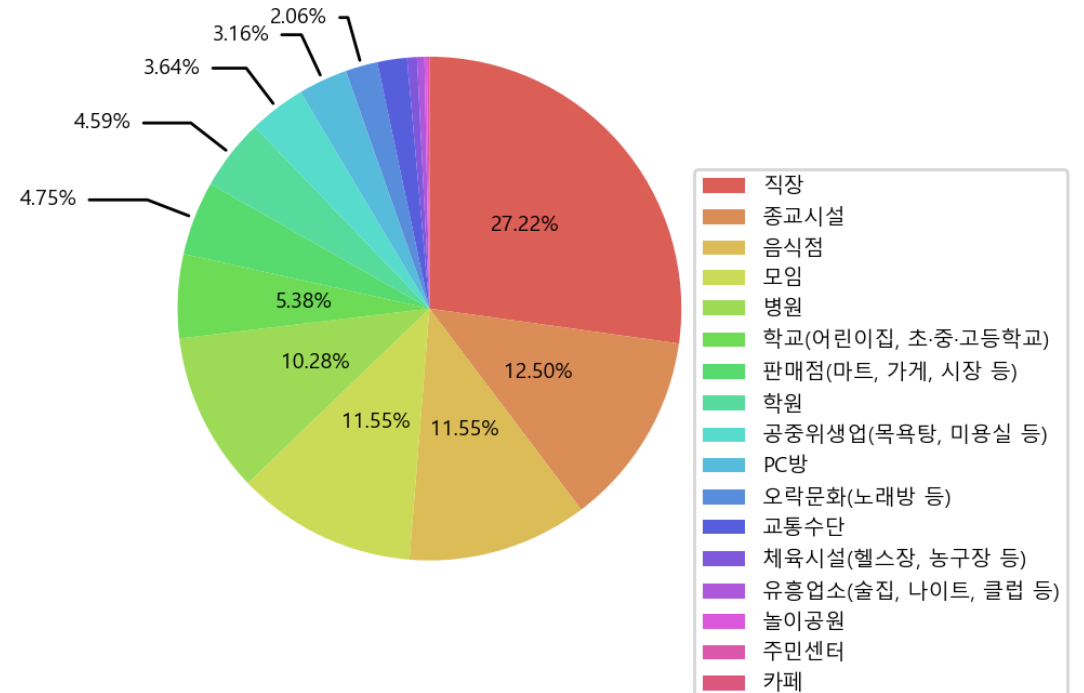
```
#파이차트 옵션
import numpy as np
import seaborn as sns

colors = sns.color_palette('hls', len(labels_frequency))

fig = plt.figure(figsize=(8,8)) ## 캔버스 크기
fig.set_facecolor('white') ## 캔버스 배경색
ax = fig.add_subplot(1,1,1) ## 프레임 생성

pie = ax.pie(frequency, ## 파이차트 출력
             startangle=90, ## 시작점을 90도(degree)
             counterclock=False, ## 시계 방향으로
             colors=colors ## 색상 지정
             )

total = np.sum(frequency) ## 빈도수 합
```



- 분류한 데이터를 가독성 좋은 시각적 통계로 구현