

# 2022년 1학기 “파이썬과학프로그래밍기초(1분반)” 프로젝트 결과 보고서

## 지역별 범죄발생율 데이터 표현

하승진 (학번:20185170)

### 1. 과제(문제) 설명

지역별 범죄발생 비율 데이터 파일을 토대로 결과를 불러와 시각화 표현하는 프로그램.

각 지역별 범죄발생비율(1000명당 1건), 범죄발생 건수, 등록인구 등의 레이블을 통해 검색, 순위, 그래프 등으로 시각적으로 보여준다.

### 2. 개발(수행)한 내용

#### [1] 라이브러리

csv(파일을 읽어오기 위함)

numpy(불러온 데이터를 행렬로 가져오기 위함)

matplotlib(그래프로 표현하기 위함)

#### [2] 구현 알고리즘

(1) 컴파일 시 프로그램 구동 전 라이브러리, 리스트 초기화, 파일을 읽은 후 dataList리스트에 저장 및 npData에 ndarray

(2) while 반복문을 통한 입력 값 'y'와 'n'를 통해 프로그램 시작 및 종료 결정

(3) 'y'로 시작 시 검색, 순위, 그래프 3가지의 선택 메뉴 출력 및 입력

(4) 정수 입력 값에 따른 각 메뉴의 기능들 구현

(5) 각 기능들은 한 Cycle로 표현까지의 단계 완료시 처음 시작단계로 넘어감

#### [3] 필요 리스트 초기화

```
dataList = []          #읽어온 데이터를 한줄 씩 저장하기 위한 리스트 초기화
calNum = [13, 10, 7, 4, 1] #년도별 데이터들을 가져오기 위한 위치지정 리스트 초기화
dataPlot = []          #[그래프]기능의 구현하기위한 데이터 저장 리스트 초기화
searchList = []         #[검색]기능의 검색하고자 하는 데이터 저장 리스트 초기화
dic_level = {}          #[순위]기능의 key를 레이블, value를 data로 저장하는 딕셔너리
```

#### [4] 파일 불러오기

```
f = open("전국_범죄발생건수_시도__fver.csv", 'r', encoding = 'CP949')
data = csv.reader(f)
for d in data:
    dataList.append(d)
npData = np.array(dataList)
# f의 open함수를 통한 파일을 불러와 읽기
```

```
# data라는 csv 객체 생성
# 반복문을 통해 data의 한줄 씩 dataList에 저장
# npData라는 행렬에 dataList의 ndarray 저장
```

## [5] 각 기능 구현

### (1) 검색

```
print("검색 지역에 대한 정보 입니다.")
name = input("지역이름 입력(영어) : ")
nyear = int(input("년도 입력 : "))
```

```
number = 2020 - nyear
```

```
for x in dataList :
    if name in x :
        for y in x :
            searchList.append(y)
```

```
print(name, "지역의", nyear, "년도 정보 입니다.")
print("범죄발생비율 :", searchList[calNum[number]], "(%) / 범죄발생 건수 :",
searchList[calNum[number]+ 1], "(건) / 등록 인구 :", searchList[calNum[number]+ 2], "명")
```

```
del searchList[:]
# 기본적으로 데이터들을 보면 각 년도별 범죄발생비율 데이터 위치는 1, 4, 7, 10, 13인덱스에 위치
# 지역(영어)이름과 년도 입력
# 반복문을 통해 줄 마다 저장된 리스트에 입력된 문자열 변수 name이 있는지 체크
# 있으면 반복문을 통해 그 줄 데이터들을 searchList리스트에 저장
# 2020값 기준으로 입력값을 빼면 2016부터 2020으로 총 4, 3, 2, 1, 0으로 흘러가기에
    calNum리스트에 뺀 값에 맞게 데이터들의 범죄발생비율 있는 행의 위치를 지정한 것을 인덱스로 넣음
Ex)2016? 2020 - 2016 = 4 , calNum[4] = 1 , searchList[1] = 2016년의 범죄발생비율
# 실행 후 searchList는 다시 프로그램 돌 경우를 생각해 del 이용해 데이터 삭제
```

### (2) 순위

```
number2 = 2020 - yNum
```

```
dic_label = npData[2:, 0].tolist()
dic_data = npData[2:, calNum[number2]].tolist()
```

```
for i in range(len(dataList)-2) :
    dic_level[dic_label[i]] = dic_data[i]
```

```
sort_level = sorted(dic_level.items(), key = lambda x : x[1], reverse = True)
```

```
print("지역별 {0}년도 기준 1위부터 {1}위 까지 정보 입니다.".format(yNum, leNum))
```

```

for i in range(leNum) :
    print("[",i+ 1,"]위 : 지역 -", sort_level[i][0], "Wt/Wt범죄발생비율 -", sort_level[i][1])
# [검색]에서의 위치 지정 리스트(calNum)활요 방식은 동일
# npData의 지역이름 데이터가 들어있는 열을 추출해 리스트로 변환 후 dic_label에 저장
# npData의 범죄발생비율 데이터가 들어있는 열을 추출해 같은 방식으로 dic_data에 저장
# 추출해 저장한 dic_label과 dic_data리스트를 반복문을 통해 dic_level 딕셔너리 객체에 저장
# sorted 함수로 dic_level의 items()로 반환된 튜플을 이용하고, lambda x[1] 즉, value를 기준으로
reverse = True, 내림차순 정렬을 하게 된다.
# 내림차순으로 정렬되어 저장된 sort_level 튜플은 반복문을 통해 출력

```

### (3) 그래프

#### - 지역

```
name = input("지역 이름 입력(영어) : ")
```

```
count = 0
```

```

for j in range(len(dataList)) :
    if name in npData[j, :] :
        plotArray = np.array(npData[j, :])
for k in range(1, len(plotArray), 3) :
    dataPlot.insert(count, float(plotArray[k]))
    count = count + 1

```

```
print(name, "의 정보 입니다.")
```

```
plt.title("Area's CrimeRate Graphics")
```

```
plt.plot(dataPlot, 'ro:')
```

```
plt.xticks([0, 1, 2, 3, 4], ['2016', '2017', '2018', '2019', '2020'])
```

```
plt.legend(['Rate'], loc = 'upper left')
```

```
plt.ylabel("rate")
```

```
plt.grid()
```

```
plt.show()
```

```

# 읽어온 npData의 모든 행을 반복문을 통해 읽으며 입력값 name이 있는 행을 plotArray에 ndarray
# 인덱스 1부터 plotArray길이만큼 까지 3씩 증가하는 k반복문을 통해 plotArray의 범죄발생비율 있는
데이터만 추출해 dataPlot리스트에 저장(why? insert는 실행 후 지우지 말고, 0번째 위치부터 같은 위
치에 덧 붙이기 식으로 저장하기 위함.
# plot을 이용한 빨강색 점선으로 표현되는 그래프 시각화

```

#### - 연도

```
number3 = 2020 - yNum
```

```
dic_data2 = []
```

```
dic_label = npData[2:, 0].tolist()
```

```
dic_data = npData[2:, calNum[number3]].tolist()
```

```

for i in dic_data :
    dic_data2.append(float(i))

plt.title("Year's Rate Graph")
plt.barh(dic_label, dic_data2, align='center', height = 0.5)
plt.xlabel('Rate')
plt.ylabel('Area')
plt.grid()
plt.show()

del dataPlot[:, dic_label[:, dic_data[:,
# [검색]에서의 위치 지정 리스트(calNum)활용 방식은 동일
# npData의 지역이름 데이터가 들어있는 열을 추출해 리스트로 변환 후 dic_label에 저장
# npData의 입력 연도의 지정 리스트 활용한 인덱스 열을 추출(즉, 연도에 해당하는 범죄발생비율 데이
터 열)해 리스트로 변환 후 dic_data에 저장
# 반복문을 통해 dic_data내용을 float타입으로 변환 후 dic_data2리스트에 저장
# ba고()함수를 이용해 레이블은 dic_label, 데이터는 dic_data2로 이루어진 수평 바 차트 그리기
# 수행 후 [그래프]기능에 사용한 dataPlot, dic_label, dic_data리스트 데이터 삭제

```

### 3. 결과 & 아쉬운 점 및 배운 점

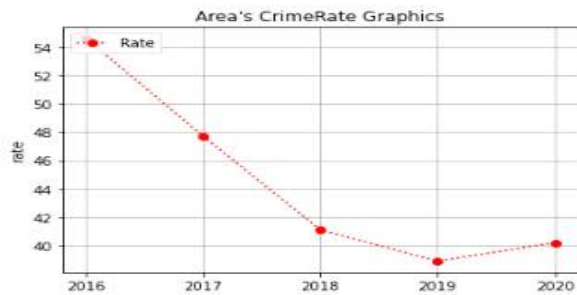
지역별 범죄발생비율(1000명당 범죄발생건수)에 대한 정보 알림 프로그램입니다.  
시작 하시겠습니까?(yes[y] / no[n]) y  
(1)검색 (2)순위 (3)그래프  
메뉴를 고르시오 : 1  
검색 지역에 대한 정보입니다.  
지역이름 입력(영어) : Gangwon  
년도 입력 : 2019  
Gangwon 지역의 2019 년도 정보입니다.  
범죄발생비율 : 29.9 (%) / 범죄발생 건수 : 46096 (건) / 등록 인구 : 1541502 명  
처음으로 돌아갑니다.

---

지역별 범죄발생비율(1000명당 범죄발생건수)에 대한 정보 알림 프로그램입니다.  
시작 하시겠습니까?(yes[y] / no[n]) y  
(1)검색 (2)순위 (3)그래프  
메뉴를 고르시오 : 2  
2016 ~ 2020년 까지의 원하는 연도와 보고싶은 범위의 순위를 입력하시오.  
년도 입력 : 2020  
몇위까지 입력 : 8  
지역별 2020년도 기준 1위부터 8위 까지 정보입니다.  
[ 1 ]위 : 지역 - Jeju / 범죄발생비율 - 40.2  
[ 2 ]위 : 지역 - Busan / 범죄발생비율 - 33.5  
[ 3 ]위 : 지역 - Seoul / 범죄발생비율 - 30.6  
[ 4 ]위 : 지역 - Daejeon / 범죄발생비율 - 30.5  
[ 5 ]위 : 지역 - Incheon / 범죄발생비율 - 30  
[ 6 ]위 : 지역 - Gwangju / 범죄발생비율 - 30  
[ 7 ]위 : 지역 - Daegu / 범죄발생비율 - 29.9  
[ 8 ]위 : 지역 - Chung-Buk / 범죄발생비율 - 29.9  
처음으로 돌아갑니다.

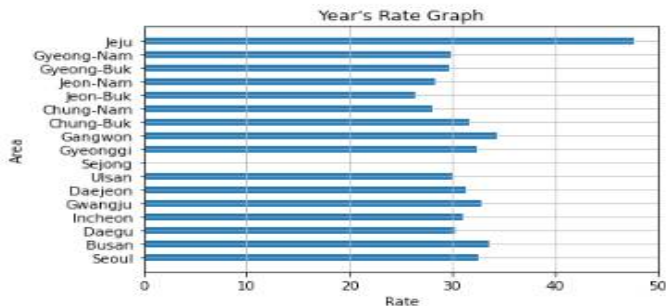
---

지역별 범죄발생비율(1000명당 범죄발생건수)에 대한 정보 알림 프로그램입니다.  
 시작 하시겠습니까?(yes[y] / no[n]) y  
 (1)검색 (2)순위 (3)그래프  
 메뉴를 고르시오 : 3  
 지역별 / 연도별범죄발생비율 시각화 정보 입니다.  
 지역별(1) / 연도별(2) 선택입력 :1  
 지역 이름 입력(영어) : Jeju  
 Jeju 의 정보 입니다.



처음으로 돌아갑니다.

지역별 범죄발생비율(1000명당 범죄발생건수)에 대한 정보 알림 프로그램입니다.  
 시작 하시겠습니까?(yes[y] / no[n]) y  
 (1)검색 (2)순위 (3)그래프  
 메뉴를 고르시오 : 3  
 지역별 / 연도별범죄발생비율 시각화 정보 입니다.  
 지역별(1) / 연도별(2) 선택입력 :2  
 연도 입력 : 2017



처음으로 돌아갑니다.

지역별 범죄발생비율(1000명당 범죄발생건수)에 대한 정보 알림 프로그램입니다.  
 시작 하시겠습니까?(yes[y] / no[n]) n  
 시스템을 종료합니다.

#### - 아쉬운 점

- \* 위치 지정 리스트를 사용하는 것이 아닌 등차수열 이용해 구현 못한 것
- \* 파일의 지역이름은 한글로 되어 있지만 jupyter의 한글 지원이 되어 있지 않아 지역 이름을 영어로 수정해 프로그램 구현한 것(구글링을 통해 font\_manager 활용으로 한글 표시해보려 했지만 실력이 모자라 끝내 못하게 되었다.)
- \* 클래스 활용으로 인한 전체 알고리즘을 완벽히 구현 못한 것

#### - 배운 점

- \* numpy행렬, list에 따른 구현으로 인해 개념에 대해 학습
- \* 파일을 읽고, 위치에 따른 데이터 추출 구현에 대해 학습
- \* 여러 그래프 시각화 구현에 따른 matplotlib 라이브러리에 대한 학습

#### 4. 참고문헌 (미주)

<공공데이터‘전국범죄발생비율’>(KOSIS통계청)

:

[https://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=DT\\_1YL3001&conn\\_path=I2](https://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=DT_1YL3001&conn_path=I2)