

알고리즘 응용(00) Lab01

201802161 조은빈

1. Goal

- 행정구역별 데이터를 취합하여 그래프 그리기
- 취합된 데이터의 총합, 평균, 분산 구하기

2. 코드 분석

- 가상 환경 생성
아래와 같이 명령어를 입력해 가상환경을 생성한다.

```
conda create -n 가상환경이름 python=3.7
```

```
(base) C:\Users\ChoEunBin>conda create -n bene python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\ChoEunBin\conda\envs\bene

  added / updated specs:
    - python=3.7

The following NEW packages will be INSTALLED:

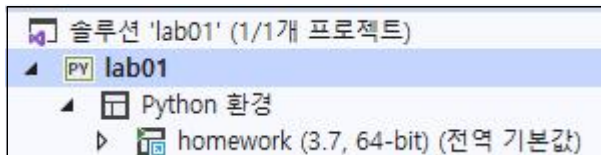
  ca-certificates pkgs/main/win-64::ca-certificates-2020.1.1-0
  certifi         pkgs/main/win-64::certifi-2019.11.28-py37_0
  openssl         pkgs/main/win-64::openssl-1.1.1d-he774522_4
  pip             pkgs/main/win-64::pip-20.0.2-py37_1
  python          pkgs/main/win-64::python-3.7.6-h60c2a47_2
  setuptools      pkgs/main/win-64::setuptools-45.0.0-py37_0
  sqlite          pkgs/main/win-64::sqlite-3.31.1-he774522_0
  vc              pkgs/main/win-64::vc-14.1-h0510ff6_4
  vs2015_runtime  pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_1
  wheel           pkgs/main/win-64::wheel-0.34.2-py37_0
  wincertstore    pkgs/main/win-64::wincertstore-0.2-py37_0

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate bene
#
# To deactivate an active environment, use
#
#   $ conda deactivate
```

다음은 통해 가상환경이 생성된 것을 확인할 수 있다.

```
(base) C:\Users\ChoEunBin>conda info --envs
# conda environments:
#
base                  *  C:\ProgramData\Anaconda3
bene                  C:\Users\ChoEunBin\conda\envs\bene
homework              C:\Users\ChoEunBin\conda\envs\homework
```

과제에서는 'homework'라는 가상환경을 생성해 사용하였다.



```
2 import numpy as np
3 import matplotlib.pyplot as plt
```

사용할 라이브러리를 import 해준다.

```
5 f = open("C:/Users/ChoEunBin/Desktop/seoul.txt", 'r', encoding = "utf-8") # 파일 불러오기
6 f.readline() # 한 줄 읽어서 인원 수가 저장된 줄로 넘어가기
```

open() 함수를 통해 텍스트 파일을 불러온다. 텍스트 파일에 저장된 데이터 양식은 다음과 같다.

seoul - Windows 메모장		파일(F)	편집(E)	서식(O)	보기(V)	도움말														
행정구역별	성별	0세	1세	2세	3세	4세	5세	6세	7세	8세	9세	10세	11세	12세	13세	14세				
종로구 계		609	590	702	765	874	877	862	1000	917	1015	927	1012	1167	1056	991	1134			
종로구 남자		311	312	334	385	438	442	431	514	460	516	486	508	585	535	503	593			
종로구 여자		298	278	368	380	436	435	431	486	457	499	441	504	582	521	488	541			
중구 계		698	659	724	813	799	766	765	771	714	680	636	661	697	582	578	691			
중구 남자		379	320	361	411	407	386	393	400	357	344	325	319	346	304	284	357			
중구 여자		319	339	363	402	392	380	372	371	357	336	311	342	351	278	294	334			

첫 번째 줄은 각 요소의 이름이 저장되어 있고 두 번째 줄부터 취합해야하는 데이터가 저장되어 있다. 때문에 readline() 함수를 통해 한 줄을 읽어서 넘긴 다음 두 번째 줄부터 데이터를 취합한다.

```

8  resultArr = np.zeros((3,101), dtype = int) # 결과 저장하는 배열
9  x = 0 # 줄 수 카운트하기 위한 변수
10
11 while True: # 파일이 끝날 때까지 한 줄씩 읽어들인다
12     x += 1
13     line = f.readline()
14     if line == "":
15         break
16
17     if x == 1: # x가 1이면 나이 별 '계' 인원수가 저장된 줄
18         totalList = line.split() # 탭을 기준으로 문자열을 나눠준다
19         totalList = totalList[2:] # 행정구역명, 성별 구분 단어 제외
20         totalArr = np.array(totalList, dtype = int) # resultArr와의 연산을 위해 resultArr와 동일한 데이터 타입 지정해서 배열 생성
21         resultArr[0] = totalArr + resultArr[0] # resultArr의 첫 번째 행에 더해 모든 행정 구역의 계 수치를 취합한다
22
23     if x == 2: # x가 2이면 나이 별 '남자' 인원수가 저장된 줄
24         maleList = line.split()
25         maleList = maleList[2:]
26         maleArr = np.array(maleList, dtype = int)
27         resultArr[1] = maleArr + resultArr[1] # resultArr의 두 번째 행에 더해 모든 행정 구역의 남자 수치를 취합한다
28
29     if x == 3: # x가 3이면 나이 별 '여자' 인원수가 저장된 줄
30         femaleList = line.split()
31         femaleList = femaleList[2:]
32         femaleArr = np.array(femaleList, dtype = int)
33         resultArr[2] = femaleArr + resultArr[2] # resultArr의 세 번째 행에 더해 모든 행정 구역의 여자 수치를 취합한다
34         x=0 # 세 번째 줄까지 읽었다면 x를 0으로 되돌려 다시 카운트
35
36 f.close()

```

모두 취합한 데이터를 저장하기 위한 빈 array를 만든다. zeros() 함수를 통해 모든 요소가 0으로 채워진 resultArr를 만든다. 저장해야할 데이터는 행에 계, 남자, 여자 3개의 데이터, 열에 0세 ~ 100세 이상까지 101개의 데이터이므로 크기는 3x101이다.

파일은 계, 남자, 여자가 차례대로 저장되어 있으므로 몇 번째 줄인지 카운트해서 구분한다. 카운트하기 위해 x라는 변수를 0으로 둔다.

무한반복문 안에서 readline()을 통해 파일을 한 줄 씩 읽는데 읽어 들인 줄이 비어있을 경우, 즉 파일을 끝까지 읽었을 경우 반복문을 종료한다. 데이터가 담겨있는 줄을 읽었을 경우 아래 코드를 실행한다.

만약 x가 1이라면 첫 번째 줄, 즉 계, 남자, 여자 중 '계'에 해당하는 데이터가 담긴 줄이다. 따라서 resultArr의 첫 번째 행에 더해줄 것이다. 우선 split() 함수를 통해 읽어온 줄 line을 탭을 기준으로 나눠준다. split()은 괄호 안의 값을 구분자로 해서 문자열을 나눠주는 함수다. 나눈 값을 totalList라는 리스트에 저장한다. totalList의 첫 번째, 두 번째 요소는 각각 행정구역명과 성별 구분이 저장되어 있어 필요 없는 부분이다. [2:]로 슬라이싱해서 세 번째 요소부터 끝까지 totalList에 다시 저장한다. 그러면 totalList에는 나이별 '계' 수치들만 저장되어 있다. resultArr에 더해주기 위해 array() 함수를 통해 totalArr로 만들어 준다. 이때 resultArr와 동일한 데이터 타입인 int로 설정한다. totalArr와 resultArr[0]을 더해서 resultArr[0]에 저장함으로써 계 수치를 취합한다. numpy를 이용해 요소간의 덧셈이 수월하다. 이 과정을 반복하면 resultArr의 첫 번째 행에는 나이별 계 수치가 취합된다.

만약 x가 2이면 두 번째 줄, 즉 계, 남자, 여자 중 '남자'에 해당하는 데이터가 담긴 줄이다. 이 데이터를 maleArr로 가공하는 과정은 위와 동일하다. maleArr와 resultArr[1]를 더해서 resultArr[1]에 저장함으로써 남자 수치를 취합한다.

만약 x가 3이면 두 번째 줄, 즉 계, 남자, 여자 중 '여자'에 해당하는 데이터가 담긴 줄이다. 이 데이터를 femaleArr로 가공하는 과정은 위와 동일하다. femaleArr와 resultArr[2]를 더해서 resultArr[2]에 저장함으로써 여자 수치를 취합한다. 세 번째 줄까지 읽었다면 x를 0으로 되돌려 다시 카운트하도록 한다.

```

38 print('계 : ', " ".join(map(str, resultArr[0]))) # resultArr의 첫 번째 행 출력
39 print('계 총합 : ', np.sum(resultArr[0], axis = 0)) # 행의 모든 요소의 총합
40 print('계 평균 : ', np.mean(resultArr[0], axis = 0, dtype = int)) # 행의 모든 요소의 평균
41 print('계 분산 : ', int(np.var(resultArr[0], axis = 0)), '\n') # 행의 모든 요소의 분산
42
43 print('남자 : ', " ".join(map(str, resultArr[1]))) # resultArr의 두 번째 행 출력
44 print('남자 총합 : ', np.sum(resultArr[1], axis = 0))
45 print('남자 평균 : ', np.mean(resultArr[1], axis = 0, dtype = int))
46 print('남자 분산 : ', int(np.var(resultArr[1], axis = 0)), '\n')
47
48 print('여자 : ', " ".join(map(str, resultArr[2]))) # resultArr의 세 번째 행 출력
49 print('여자 총합 : ', np.sum(resultArr[2], axis = 0))
50 print('여자 평균 : ', np.mean(resultArr[2], axis = 0, dtype = int))
51 print('여자 분산 : ', int(np.var(resultArr[2], axis = 0)), '\n')

```

resultArr에 모든 행정구역의 데이터가 취합되었으므로 이를 출력한다. 데이터를 출력하기 위해 사용한 함수는 join()이다. join()함수의 파라미터로 문자열을 입력하면 문자열 사이에 특정 문자를 삽입하여 새로운 문자열로 합치는 기능을 한다. 여기서는 구분자로 " "(공백)을 사용하여 나이별 인원수에 대한 데이터들을 공백으로 구분되는 하나의 문자열로 합치고 이를 출력하도록 했다. 또 join 함수의 파라미터 형식에 맞추기 위해 map() 함수를 이용해 resultArr[0]의 형태를 문자열로 바꿔주었다.

총합을 계산하기 위해서는 sum() 함수를 사용하였다. 각각의 행에 있는 요소들끼리만 더해줄 것이므로 axis = 0으로 설정한다. axis = 0은 x축(행)을 기준으로 합을 구하는 방식이다. axis 설정을 해주지 않으면 함수는 모든 요소의 합을 반환한다.

평균을 계산하기 위해서는 mean() 함수를 사용하였다. sum() 함수를 사용했을 때처럼 axis = 0으로 설정해주고 dtype(데이터 타입)을 int로 설정한다. 데이터 타입을 따로 설정해주지 않으면 기본 자료형인 float으로 결과가 반환된다.

분산을 계산하기 위해서는 var() 함수를 사용하였다. sum() 함수를 사용했을 때처럼 axis = 0으로 설정해 준다. 이때에는 dtype 설정을 미리 해주지 않고 나중에 int로 형 변환을 한다. 그 이유는 var() 함수의 옵션으로 dtype 설정을 미리 int로 할 경우 계산 결과가 int의 범위를 넘어서면 음수가 반환된다. 따라서 float 형으로 계산한 다음 int로 형 변환하면 원하는 결과를 반환할 수 있다.

```

53 plt.figure(figsize=(16,4)) # 최초 창 크기 설정
54
55 plt.subplot(131) # 하나의 창에 그래프 여러 개 그릴 수 있도록 구역 지정
56 plt.bar(np.arange(101), resultArr[0])
57 plt.title('Total')
58
59 plt.subplot(132)
60 plt.bar(np.arange(101), resultArr[1])
61 plt.title('Male')
62
63 plt.subplot(133)
64 plt.bar(np.arange(101), resultArr[2])
65 plt.title('Female')
66
67 plt.show()

```

figsize를 임의로 설정해서 그래프를 보기 쉽도록 최초의 창 크기를 조절한다.

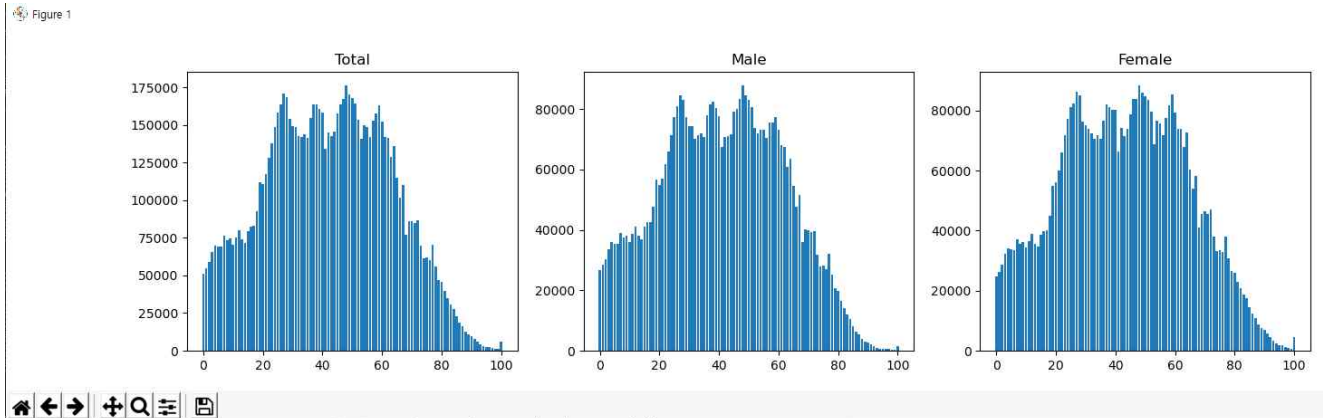
subplot() 함수를 통해 그래프의 좌표를 설정한다. 총 3개의 그래프를 그릴 것이기 때문에 1x3 형태로 배치한다.

그래프를 그리기 위해 사용한 함수는 bar()이다. bar() 함수는 막대그래프를 그리는 함수이다. 첫 번째 인

자는 x축 값, 두 번째 인자는 y축 값이다. x축에는 0세부터 100세까지 즉, 0부터 100까지의 연속된 수를 입력하기 위해 `arange()` 함수를 사용했다. y축 값으로는 각각 계, 남자, 여자의 나이별 인원수에 대한 데이터를 입력한다.

3. 실행 결과

- Goal 1



- Goal 2

```
C:\Users\ChoEunBin\#conda\envs\#homework\#python.exe
계 : 51145 54779 59128 65734 70013 69139 69031 76160 73259 74305 70378 74930 79942 73725 71582 79677 82287 82767 92621
111662 110727 117160 127846 137538 148693 158278 163318 170959 168286 153818 149445 148368 142549 141939 143557 141456 1
54667 163441 163787 160750 157941 133933 145000 142609 145629 157862 163865 167243 176438 170463 167963 164022 153253 14
0798 149743 148790 142141 153039 157352 162867 152404 141959 141460 128754 136069 114979 101482 110044 76918 85733 86235
84893 86715 69827 61120 61736 59938 70345 56062 47216 45815 39499 34909 30674 27939 22645 18612 16033 12629 10732 9536
7732 6115 4258 3207 2451 2420 1753 1241 977 6244
계 총합 : 9729107
계 평균 : 96327
계 분산 : 3053116216

남자 : 26523 28466 30328 33582 35843 35417 35449 39091 37504 38131 35988 38555 41149 38075 36993 41112 42430 42658 4762
5 56754 54727 57065 61644 65825 71292 77215 80801 84616 83128 77347 74220 74290 70092 71242 71802 70789 77892 81497 8254
7 80459 77745 67495 70749 71127 71669 79030 80110 83328 87909 84366 83115 80531 73690 71947 73166 73078 70278 75544 7566
8 77425 72978 68057 67501 60851 63431 54594 47566 51668 35992 40158 39849 39382 39511 31763 27992 28184 27100 32181 2522
1 20707 19728 16558 14035 11927 10459 8039 6373 5257 3815 3040 2735 2022 1525 1002 724 643 677 450 337 298 1596
남자 총합 : 4744059
남자 평균 : 46970
남자 분산 : 772066863

여자 : 24622 26313 28800 32152 34170 33722 33582 37069 35755 36174 34390 36375 38793 35650 34589 38565 39857 40109 4499
6 54908 56000 60095 66202 71713 77401 81063 82517 86343 85158 76471 75225 74078 72457 70697 71755 70667 76775 81944 8124
0 80291 80196 66438 74251 71482 73960 78832 83755 83915 88529 86097 84848 83491 79563 68851 76577 75712 71863 77495 8168
4 85442 79426 73902 73959 67903 72638 60385 53916 58376 40926 45575 46386 45511 47204 38064 33128 33552 32838 38164 3084
1 26509 26087 22941 20874 18747 17480 14606 12239 10776 8814 7692 6801 5710 4590 3256 2483 1808 1743 1303 904 679 4648
여자 총합 : 4985048
여자 평균 : 49356
여자 분산 : 759968806
```