



전자공학과 2020142001 곽종근



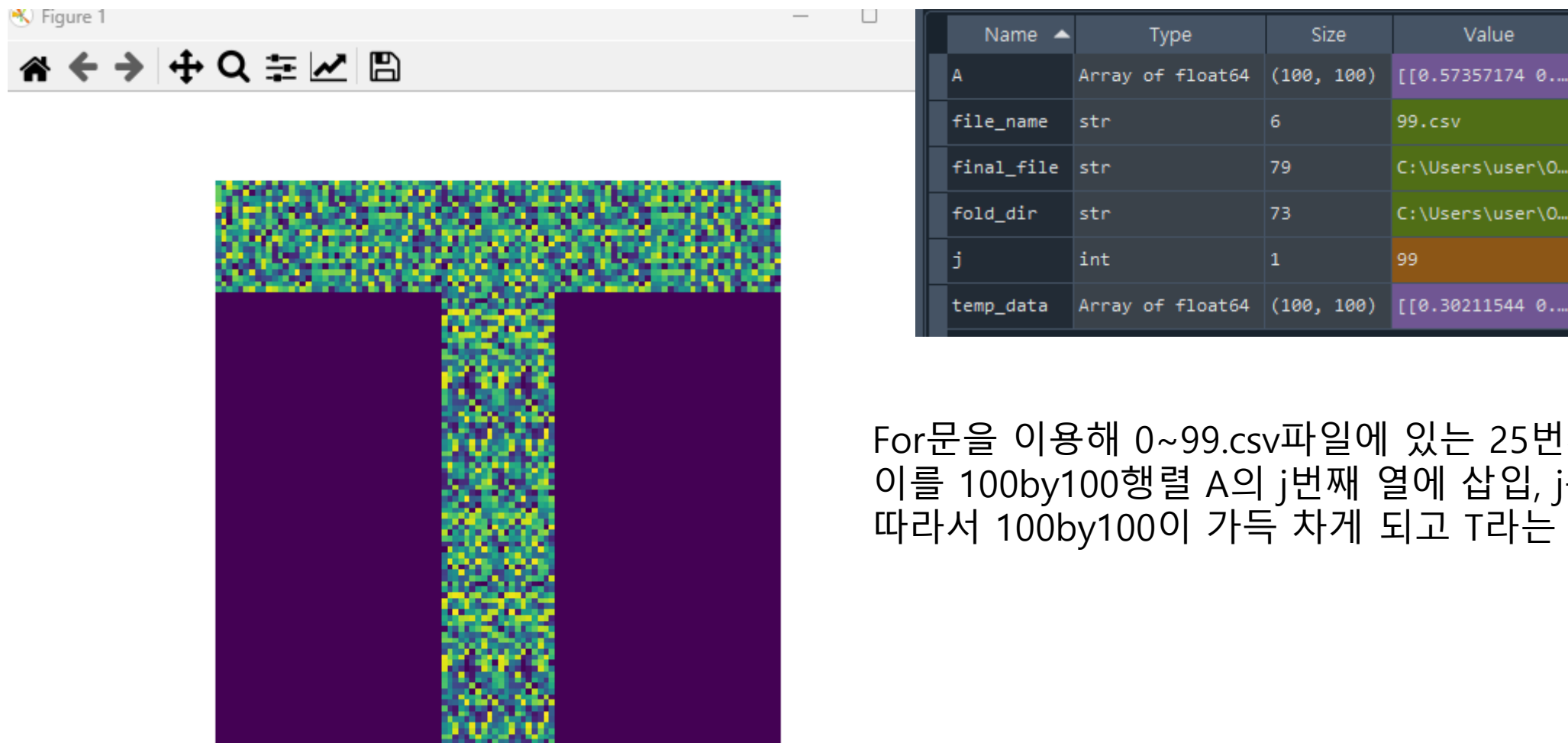
데이터 처리 실습

머신러닝실습(01)

과제 1-1. T 출력(코드)

```
1 import numpy as np      #numpy를 np라는 이름으로 사용하겠다.
2 import pandas as pd     #pandas를 pd라는 이름으로 사용하겠다.
3 import matplotlib.pyplot as plt    #matplotlib.pyplot를 plt라는 이름으로 사용하겠다.
4
5 A=np.zeros([100,100])   #A라는 100by100의 zeros배열을 선언해놓는다.
6
7
8
9
10 for j in range(0,100,1):    #파일이 총 100개가 있기때문에 이 100개의 파일을 모두 사용하기 위해 0~99까지 for문(반복문)을 이용해 전부 선언, 오름차순이므로 0부터
11 fold_dir="C:\\Users\\user\\OneDrive - 한국공학대학교\\바탕 화면\\3학년 1학기\\머신러닝실습\\4주차\\problem_1_data\\"
12 file_name=str(j)+".csv"    #반복문 j를 j.csv로 선언하여 0~99.csv가 전부 불러와질 수 있도록 선언한다.
13 final_file=fold_dir+file_name    #fold_dir과 file_name을 합쳐 파일이 해당하는 위치의 0~99개 파일을 불러오는 final_str을 최종 선언
14
15
16 temp_data=pd.read_csv(final_file, header=None) #final_file이라는 최종 파일에 있는 데이터를 읽어오는데 헤더파일을 생성하지 않고 읽어온다. dataframe형태
17 temp_data=temp_data.to_numpy() #위에서 불러온 파일은 dataframe형태이므로 슬라이싱이 안되기에 이를 numpy배열로 변환시킨다.
18 A[:,j]=temp_data[:,25] #위에서 numpy배열로 변환해주었기때문에 슬라이싱이 가능하고 dataframe형태는 불가능하다.
19 #25번째 col을 행렬 A의 0번부터 99번까지 col에 삽입한다.
20
21
22
23 plt.imshow(A, cmap='viridis')
24 plt.axis('off')
25 plt.show
26
27
```

과제 1-1. T 출력(출력)

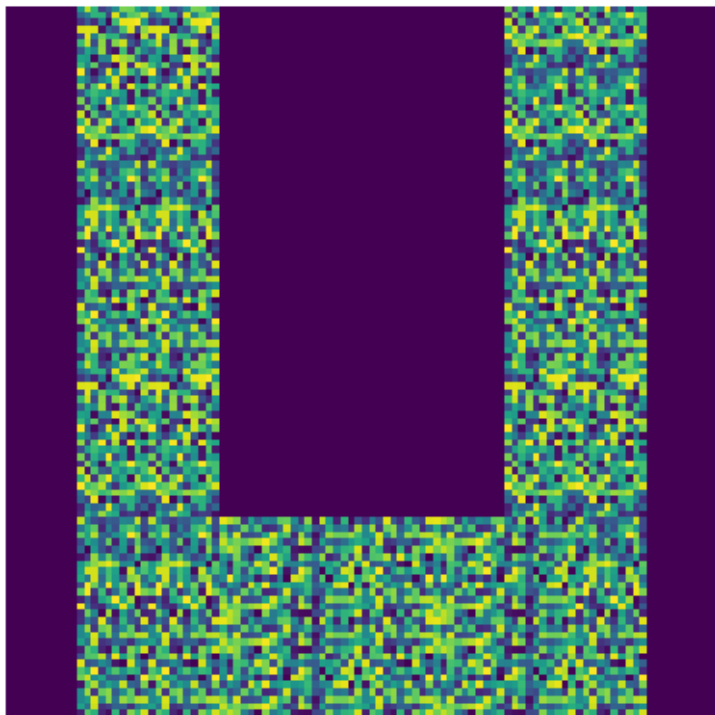
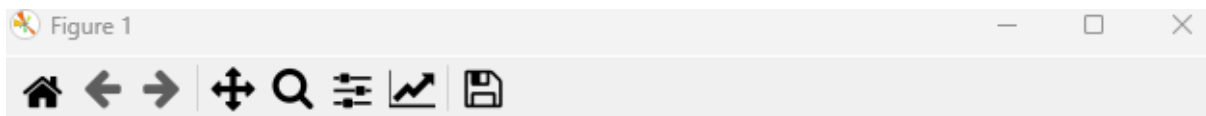


For문을 이용해 0~99.csv파일에 있는 25번째 column을 추출
이를 100by100행렬 A의 j번째 열에 삽입, j는 0~99번 까지
따라서 100by100이 가득 차게 되고 T라는 문자가 나타난다

과제 1-2. U 출력(코드)

```
1 import numpy as np #numpy를 np라는 이름으로 사용하겠다.
2 import pandas as pd #pandas를 pd라는 이름으로 사용하겠다.
3 import matplotlib.pyplot as plt #matplotlib.pyplot를 plt라는 이름으로 사용하겠다.
4
5 B=np.zeros([100,100]) #B라는 100by100의 zeros배열을 선언해놓는다.
6
7
8
9
10
11 for j in range(0,100,1): #파일이 총 100개가 있기때문에 이 100개의 파일을 모두 사용하기 위해 0~99까지 for문(반복문)을 이용해 전부 선언, 오름차순이므로 0부터
12 fold_dir="C:\\Users\\user\\OneDrive - 한국공학대학교\\바탕 화면\\3학년 1학기\\머신러닝실습\\4주차\\problem_1_data\\"
13 file_name=str(j)+".csv" #반복문 j를 j.csv로 선언하여 0~99.csv가 전부 불러와질 수 있도록 선언한다.
14 final_file=fold_dir+file_name #fold_dir과 file_name을 합쳐 파일이 해당하는 위치의 0~99개 파일을 불러오는 final_str을 최종 선언
15
16
17 temp_data=pd.read_csv(final_file, header=None) #final_file이라는 최종 파일에 있는 데이터를 읽어오는데 헤더파일을 생성하지 않고 읽어온다. dataframe형태
18 temp_data=temp_data.to_numpy() #위에서 불러온 파일은 dataframe형태이므로 슬라이싱이 안되기에 이를 numpy배열로 변환시킨다.
19 B[j,:]=temp_data[10,:] #위에서 numpy배열로 변환해주었기때문에 슬라이싱이 가능하고 dataframe형태는 불가능하다.
20 #25번째 col을 행렬 A의 0번부터 99번까지 col에 삽입한다.
21
22
23
24 plt.imshow(B, cmap='viridis')
25 plt.axis('off')
26 plt.show
27
```

과제 1-2. U 출력(출력)



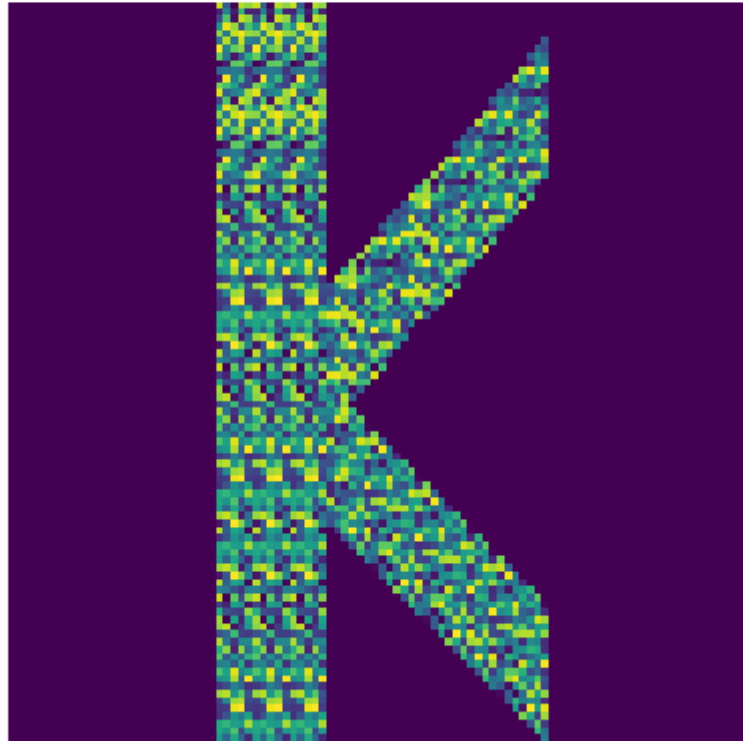
Name	Type	Size	Value
B	Array of float64	(100, 100)	[[0. 0. 0. ... [0. 0. 0. ...
file_name	str	6	99.csv
final_file	str	79	C:\Users\user\O...
fold_dir	str	73	C:\Users\user\O...
j	int	1	99
temp_data	Array of float64	(100, 100)	[[0.30211544 0...

For문을 이용해 0~99.csv파일에 있는 10번째 행을 추출
이를 100by100행렬 A의 j번째 행에 삽입, j는 0~99번 까지
따라서 100by100이 가득 차게 되고 U라는 문자가 나타난다

과제 1-3. K 출력(출력)

```
1 import numpy as np          #numpy를 np라는 이름으로 사용하겠다.
2 import pandas as pd         #pandas를 pd라는 이름으로 사용하겠다.
3 import matplotlib.pyplot as plt #matplotlib.pyplot를 plt라는 이름으로 사용하겠다.
4
5 C=np.zeros([100,100]) #C라는 100by100의 zeros배열을 선언해놓는다.
6
7
8
9
10
11 for j in range(0,10,1):      #T와 U의 출력에서는 행 또는 열만 바뀌었는데 K의 출력은 각 행과 열에 해당하는 값을 넣어주어야 하므로 이중배열로 만들어 주기 위해
12                               #이중for문을 만들어 준다. j가 먼저 실행되고 그 뒤 i가 0~9까지 실행이 된 후 j가 1로 넘어가는 순서이므로 j를 10의자리 수, i를 1의자리 수 라고
13                               #판단이 가능하다. 따라서 아래와 같은 for문을 작성하였다.
14
15     for i in range(0,10,1):
16         fold_dir="C:\\Users\\user\\OneDrive - 한국공학대학교\\바탕 화면\\3학년 1학기\\머신러닝실습\\4주차\\problem_1_data\\"
17         file_name=str(i+j*10)+".csv" #j에 10을 곱해주면 j는 i가 0~9까지 변할 때 j는 0, 10, 20 ~ 90의 순서로 변화하므로 이는 0~99까지 변화하고 모든 파일을 전부 반복문으로 입력받을 수 있다.
18         final_file=fold_dir+file_name #fold_dir과 file_name을 합쳐 파일이 해당하는 위치의 0~99개 파일을 불러오는 final_str을 최종 선언
19         temp_data=pd.read_csv(final_file, header=None) #final_file이라는 최종 파일에 있는 데이터를 읽어오는데 헤더파일을 생성하지 않고 읽어온다. dataframe형태로 읽어와진다.
20         temp_data=temp_data.to_numpy() #위에서 불러온 파일은 dataframe형태이므로 슬라이싱이 안되기에 이를 numpy배열로 변환시킨다.
21         C[j*10:(j*10)+10,i*10:(i*10)+10]=temp_data[70:80,80:90] #위에서 numpy배열로 변환해주었기때문에 슬라이싱이 가능하고 dataframe형태는 불가능하다.
22
23
24 #C에 temp_data의 70~79, 80~89번째의 데이터를 10by10으로 C에 넣어줘야 하는데 이를 위해 직접 배열을 나눠 보았더니
25 #      0번째      1번째      2번째      3번째      4번째      5번째      6번째      7번째      8번째      9번째
26 # C[0:10, 0:10] [0:10, 10:20] [0:10, 20:30] [0:10, 30:40] [0:10, 40:50] [0:10, 50:60] [0:10, 60:70] [0:10, 70:80] [0:10, 80:90] [0:10, 90:100]
27 #      10       11       12       13       14       15       16       17       18       19
28 # C[10:20, 0:10] [10:20, 10:20] [10:20, 20:30] [10:20, 30:40] [10:20, 40:50] [10:20, 50:60] [10:20, 60:70] [10:20, 70:80] [10:20, 80:90] [10:20, 90:100]
29 # C[20:30, 0:10] [20:30, 10:20] [20:30, 20:30] [20:30, 30:40] [20:30, 40:50] [20:30, 50:60] [20:30, 60:70] [20:30, 70:80] [20:30, 80:90] [20:30, 90:100]
30 #이러한 형태의 변화가 나타나는데 여기서 보면, 일단 for문이 j먼저 선언되면 i가 0~9까지 변화 후 j가 변화하는 것을 이용하여 먼저 행 부분에서는 j*10으로 시작해서 (j*10)+10을 해 주면 각 변화하는 값에 맞춰
31 #10개의 행을 선언하게 되고, 열은 i*10으로 시작하여 (i*10)+10의 형태로 10개의 열이 되어 총 10by10의 행렬에 temp_data값을 넣어줄 수 있다. 이중for문의 특성으로 값이 밀린다고 생각하면 쉽다.
32
33
34 plt.imshow(C, cmap='viridis')
35 plt.axis('off')
36 plt.show
```

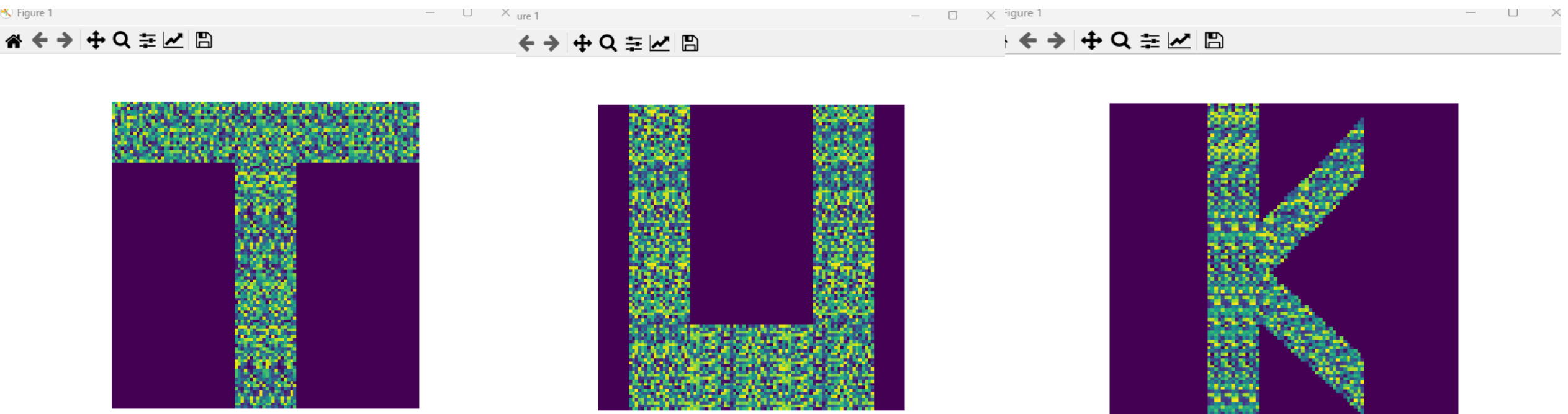
과제 1-3. K 출력(코드)



Name	Type	Size	Value
C	Array of float64	(100, 100)	[[0. 0. 0. [0. 0. 0.
file_name	str	6	99.csv
final_file	str	79	C:\Users\user\0...
fold_dir	str	73	C:\Users\user\0...
i	int	1	9
j	int	1	9
temp_data	Array of float64	(100, 100)	[[0.30211544 0. ...

For문을 이용해 0~99.csv파일에 있는 70~79번째 행
80~89번째 열을 추출하여 10by10의 데이터로 잘리게 되고
이를 100by100행렬 C에 차례대로 삽입하였고 K가 나타난다.

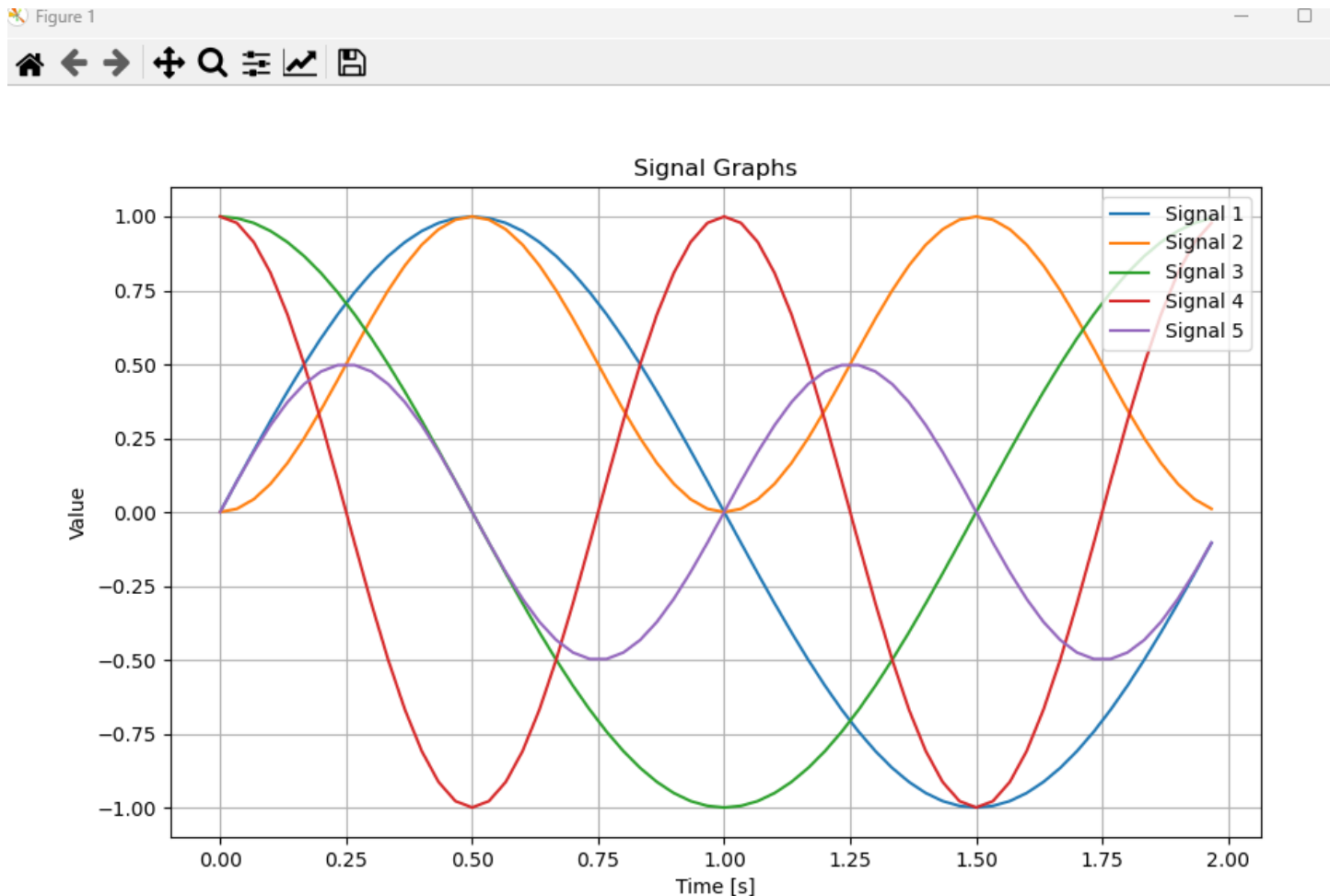
과제 1번. T U K 출력



과제 3번. 주파수 처리 코드

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5
6
7 fold_dir="C:\\Users\\user\\OneDrive - 한국공학대학교\\바탕 화면\\3학년 1학기\\머신러닝실습\\4주차\\problem_2_data.csv"
8 temp_data=pd.read_csv(fold_dir)
9
10
11 # NaN 값을 ''으로 대체하여 300x5 크기를 유지
12 new_td = temp_data.fillna('')
13
14 total_counts = np.zeros([1,5]) #값을 1by5에 저장하라고 하였으므로 1,5의 빈 배열을 선언한다. 전체 데이터 값
15 sam_counts = np.zeros([1,5]) #위와 같은 내용이고 이 부분에는 샘플링 값들만 넣어 저장
16
17
18 for i in range(5):
19     total_counts[0, i] = sum(new_td.iloc[:, i] != '') #i열에서 nan을 대체한 ''값을 제외하고 남은 데이터의 개수를 구한다
20     sam_counts[0, i] = sum(new_td.iloc[:, i] != '') // 2 #열에서 nan을 대체한 ''값을 제외하고 남은 데이터의 개수를 구한다
21                                     #샘플링한 값이고, 2초의 시간이라 하였으므로 2로 나눠준 값이 저장된다.
22
23
24
25 fmin = np.min(sam_counts) #np.min()을 이용해 괄호 안에 최소값을 구하고자 하는 배열을 넣으면 된다.
26
27 DS_data=np.zeros([300,5]) #300by5의 제로스 빈 배열을 선언해준다.
28
29
30 for i in range(5):
31     # 다운샘플링 간격 계산
32     # 현재 열의 전체 데이터 수인 total_counts를 30으로 나눈 값을 간격으로 설정
33
34     DS_interval = int(sam_counts[0, i] // fmin)
35
36     # 빈 문자열을 가진 행을 제외하고 다운샘플링하여 DS_data에 저장
37     # nan값을 ''으로 설정하였기에 먼저 데이터프레임이므로 iloc[:,i]를 이용해 현재 신호의 i번째 열 선택
38     # 그 다음 조건으로 ''을 제외한 값을 선택하고 이 선택된 부분중 다운샘플링 간격으로 샘플링을 한다. 이 모든 조건을 대괄호[]을 붙여서 사용해주면 한줄에 작성이 가능하다.
39     DS_col = new_td.iloc[:, i][new_td.iloc[:, i] != ''][::DS_interval]
40     DS_data[:len(DS_col), i] = DS_col
41
42
43 plt.figure(figsize=(10,6))
44 for i in range(5):
45     plt.plot(np.arange(0,2,1/fmin),DS_data[0:60,i])
46 plt.title("Signal Graphs")
47 plt.xlabel("Time [sec]")
48 plt.ylabel("Value [V]")
49 plt.grid(True)
50 plt.legend(["Signal1","Signal2","Signal3","Signal4","Signal5"],loc="upper right")
51
52
```

과제 3번. 주파수 처리 출력



Name	Type	Size	Value
DS_col	Series	(60,)	Series object of pandas.core.series module
DS_data	Array of float64	(300, 5)	[[0. 0. 1. ... [0.10452846 ...
DS_interval	int	1	3
fmin	float64	1	30.0
fold_dir	str	76	C:\Users\user\OneDrive - 한...
i	int	1	4
new_td	DataFrame	(300, 5)	Column names: Signal1, Signa...
sam_counts	Array of float64	(1, 5)	[[30. 120. 60. 150. 90.]]
temp_data	DataFrame	(300, 5)	Column names: Signal1, Signa...
total_counts	Array of float64	(1, 5)	[[60. 240. 120. 300. 180.]]

2번 데이터 파일에서 nan값을 "(공백)"으로 바꿔준 후 각 데이터를 dataframe형태로 불러온다.
5개의 신호에 대한 샘플링 주파수를 1by5의 배열에 저장한 후
이들 중 가장 작은 주파수를 저장하여 이 주파수가 샘플링 주파수가 되도록
다른 신호를 다운샘플링 시켜주어 그래프로 출력한다.