

[실습-1] 주요 대학별 2024학년도 수시2차 실시간 지원현황 데이터 분석하기

> 주요 대학의 '전형별 경쟁률 현황' 을 스크랩핑하여 분석한다.

0. 주요 대학의 2024학년도 수시2차 실시간 지원현황 페이지를 방문하여 페이지 정보 알아보기

부천대학교, 유한대학교

1. 전체 경쟁률 현황을 데이터 프레임으로 구축하기

대학명, 경쟁률 페이지 url, 분석 대상 table의 index 사용\

> [실습] 웹 페이지에서 원하는 \

의 데이터를 수집해주는 함수 만들어 사용

2. 추출된 데이터에 대한 정제

> 불필요한 컬럼 제거

['전형', '모집인원', '지원인원', '경쟁률']와 관련된 컬럼만 남기고 제거

> 컬럼명 통일

['전형', '모집인원', '지원인원', '경쟁률']로 통일

> '대학' 컬럼 추가

> 대학별 데이터 행들을 하나의 데이터 프레임으로 통합

> 경쟁률 데이터 값 보정하기

후반부의 ':1' 값 제거\ 자료형을 float로 변경

> 불필요한 행 제거

'전형' 컬럼에 '총계'나 '소계' 값이 들어있는 행은 삭제\ 문자열에서 찾는 문자열의 시작 위치 값 찾기 : **str.find()**

> 컬럼 순서 조정

['대학', '전형', '모집인원', '지원인원', '경쟁률'] 순으로 조정

3. [분석] 대학별 합계 경쟁률 데이터 구축

대학별 모집인원, 지원인원, 총지원인원, 경쟁률 컬럼으로 구성
\ 경쟁률이 높은 순으로 보기

4. [분석] 대학별 합계 경쟁률 비교 그래프 그리기

대학별 합계 경쟁률 순위 막대 그래프 그리기\ 대학별 합계 지원자 수 순위 막대 그래프 그리기

0. 주요 대학의 2024학년도 수시2차 실시간 지원현황 페이지를 방문하여 페이지 정보 알아보기

부천대학교, 유한대학교

```
In [80]: ## [부천대학교] URL로 웹 페이지 열어보기
import webbrowser as wb
url = 'http://ratio.uwayapply.com/SmZKTEJKZkNDYUxKemZUZg=='
wb.open_new(url) #브라우저에 사이트가 열림
```

Out[80]: True

함수 만들기 : URL로 웹 페이지 열어보기

```
In [5]: ## [함수 만들기] URL로 웹 페이지 열어보기
def open_page(url='https://www.example.com'):
    import webbrowser as wb
    wb.open_new(url) #브라우저에 사이트가 열림
```

```
In [6]: ## [함수 호출] URL로 웹 페이지 열어보기
url = 'http://ratio.uwayapply.com/SmZKTEJKZkNDYUxKemZUZg=='
open_page(url)
```

1. 전형별 경쟁률 현황을 데이터 프레임으로 구축하기

대학명, 경쟁률 페이지 url, 분석 대상 table의 index 사용

```
In [8]: ## HTTP 요청 헤더 정보 설정: 클라이언트가 서버에게 요청과 관련된 추가 정보를 제공하는 데 사용
header = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3',
    'Referer': url
}
```

[부천대학교] 해당 페이지의 \\
데이터를 받아 데이터 프레임 구축

원하는 \\

데이터만 추출

```
In [9]: ## [부천대학교] 경쟁률 현황을 데이터 프레임으로 구축하기
import pandas as pd
import urllib3

url = 'http://ratio.uwayapply.com/SmZKTEJKZkNDYUxKemZUZg=='
index = 1 # 전형별 경쟁률 현황

# 웹 페이지에서 표 스크래핑
http = urllib3.PoolManager() #http나 https 연결 관리자 호출
```

```

req = http.request('GET', url, headers=header) #http 연결 관리자
로 html 문서 요청
tables = pd.read_html(req.data) #html 문서에서 테이블 태그
(<table> </table>) 부분의 값들을 List 형태(table 개수만큼)로 반환

df_a = tables[index] #데이터 프레임
df_a

```

Out[9]:

	구분	전형	총모집인원	지원인원	경쟁률
0	정원내특별전형	일반고	488	5136	10.52 : 1
1	정원내특별전형	특성화고	127	1147	9.03 : 1
2	정원내특별전형	특기자	27	248	9.19 : 1
3	정원외특별전형	수급자 및 차상위	19	174	9.16 : 1
4	정원외특별전형	농어촌	9	37	4.11 : 1
5	정원외특별전형	서해5도	16	0	0.00 : 1
6	정원외특별전형	북한이탈주민	17	1	0.06 : 1
7	정원외특별전형	전문대 이상 졸업자	44	172	3.91 : 1
8	총계	총계	747	6915	9.26 : 1

[유한대학교] URL로 웹 페이지 열어보기

In [15]:

```

## [유한대학교] URL로 웹 페이지 열어보기
url = 'http://ratio.uwayapply.com/SmZKVzh80mJKZkNDYUxKemZUZg=='
open_page(url)

```

[유한대학교] 해당 페이지의 \ 데이터를 받아 데이터 프레임 구축

| 원하는 \

데이터만 추출

[실습] 웹 페이지에서 원하는 \

의 데이터를 수집해주는 함수 만들어 사용

| 함수로 수집할 대학의 url과 \

index 값, header 값을 넘겨서 호출함(단, header 값은 생략 가능하게)\ 함수는 url에 해당하는 페이지에서 index에 해당하는 \
값만으로 데이터 프레임을 구성해서 반환

In [10]:

```

## [함수 만들기] 경쟁률 현황을 데이터 프레임으로 구축하여 반환

```

In [18]:

```

## [유한대학교] 경쟁률 현황을 데이터 프레임으로 구축하기

```

df_b

Out[18]:

구분		모집전형	총 모집인원	지원인원	경쟁률
0	정원내	일반고 전형	126	1981	15.72 : 1
1	정원내	특성화고 전형	105	764	7.28 : 1
2	정원내 소계	정원내 소계	231	2745	11.88 : 1
3	정원외	전문대학 이상 졸업자 전형	35	18	0.51 : 1
4	정원외	기초생활수급자 및 차상위계층 전형	36	154	4.28 : 1
5	정원외	서해5도 교육과정 이수자 전형	7	0	0.00 : 1
6	정원외	외국인 및 재외국민 전형	36	2	0.06 : 1
7	정원외	재직자(특성화고 등을 졸업한 재직자) 전형	63	7	0.11 : 1
8	정원외 소계	정원외 소계	177	181	1.02 : 1
9	총계	총계	408	2926	7.17 : 1

2. 추출된 데이터에 대한 정제

불필요한 컬럼 제거

> ['전형', '모집인원', '지원인원', '경쟁률']와 관련된 컬럼만 남기고 제거

```
df = df.drop('구분', axis = 'columns')\ del df_a['구분']
```

In [129...]

```
#### [부천대학] 컬럼명 통일시키기
## ['구분', '전형', '모집인원', '지원인원', '경쟁률']와 관련된 컬럼만 남기고 제거
#df_a = df_a.drop('구분', axis = 'columns') # 컬럼 이름으로 컬럼 제거
del df_a['구분'] # 불필요한 컬럼 제거
```

```
df_a
```

Out[129]:

	전형	총모집인원	지원인원	경쟁률
0	일반고	488	5136	10.52 : 1
1	특성화고	127	1147	9.03 : 1
2	특기자	27	248	9.19 : 1
3	수급자 및 차상위	19	174	9.16 : 1
4	농어촌	9	37	4.11 : 1
5	서해5도	16	0	0.00 : 1
6	북한이탈주민	17	1	0.06 : 1
7	전문대 이상 졸업자	44	172	3.91 : 1
8	총계	747	6915	9.26 : 1

컬럼명 통일

```
['전형', '모집인원', '지원인원', '경쟁률']로 통일
```

In [130...]

```
## ['구분', '전형', '모집인원', '지원인원', '경쟁률']로 컬럼명 통일
new_cols = ['전형', '모집인원', '지원인원', '경쟁률']
df_a.columns = new_cols

df_a
```

Out[130]:

	전형	모집인원	지원인원	경쟁률
0	일반고	488	5136	10.52 : 1
1	특성화고	127	1147	9.03 : 1
2	특기자	27	248	9.19 : 1
3	수급자 및 차상위	19	174	9.16 : 1
4	농어촌	9	37	4.11 : 1
5	서해5도	16	0	0.00 : 1
6	북한이탈주민	17	1	0.06 : 1
7	전문대 이상 졸업자	44	172	3.91 : 1
8	총계	747	6915	9.26 : 1

In [131...]

```
#### [유한대학] 컬럼명 통일시키기
## ['구분', '전형', '모집인원', '지원인원', '경쟁률']와 관련된 컬럼만 남기고 제거
```

```
## ['구분', '전형', '모집인원', '지원인원', '경쟁률']로 컬럼명 통일
```

```
df_b
```

Out[131]:

	전형	모집인원	지원인원	경쟁률
0	일반고 전형	126	1981	15.72 : 1
1	특성화고 전형	105	764	7.28 : 1
2	정원내 소계	231	2745	11.88 : 1
3	전문대학 이상 졸업자 전형	35	18	0.51 : 1
4	기초생활수급자 및 차상위계층 전형	36	154	4.28 : 1
5	서해5도 교육과정 이수자 전형	7	0	0.00 : 1
6	외국인 및 재외국민 전형	36	2	0.06 : 1
7	재직자(특성화고 등을 졸업한 재직자) 전형	63	7	0.11 : 1
8	정원외 소계	177	181	1.02 : 1
9	총계	408	2926	7.17 : 1

'대학' 컬럼 추가

In [132...]

```
## [부천대학] 대학 열 추가
df_a['대학'] = '부천대학교'
df_a
```

Out[132]:

	전형	모집인원	지원인원	경쟁률	대학
0	일반고	488	5136	10.52 : 1	부천대학교
1	특성화고	127	1147	9.03 : 1	부천대학교
2	특기자	27	248	9.19 : 1	부천대학교
3	수급자 및 차상위	19	174	9.16 : 1	부천대학교
4	농어촌	9	37	4.11 : 1	부천대학교
5	서해5도	16	0	0.00 : 1	부천대학교
6	북한이탈주민	17	1	0.06 : 1	부천대학교
7	전문대 이상 졸업자	44	172	3.91 : 1	부천대학교
8	총계	747	6915	9.26 : 1	부천대학교

In [133...]

```
## [유한대학] 대학 열 추가
```

```
df_b
```

Out[133]:

	전형	모집인원	지원인원	경쟁률	대학
0	일반고 전형	126	1981	15.72 : 1	유한대학교
1	특성화고 전형	105	764	7.28 : 1	유한대학교
2	정원내 소계	231	2745	11.88 : 1	유한대학교
3	전문대학 이상 졸업자 전형	35	18	0.51 : 1	유한대학교
4	기초생활수급자 및 차상위계층 전형	36	154	4.28 : 1	유한대학교
5	서해5도 교육과정 이수자 전형	7	0	0.00 : 1	유한대학교
6	외국인 및 재외국민 전형	36	2	0.06 : 1	유한대학교
7	재직자(특성화고 등을 졸업한 재직자) 전형	63	7	0.11 : 1	유한대학교
8	정원외 소계	177	181	1.02 : 1	유한대학교
9	총계	408	2926	7.17 : 1	유한대학교

대학별 데이터 행들을 하나의 데이터 프레임으로 통합

In [134...]

```
## 대학별 데이터 행 통합하기 : index는 무시
import pandas as pd
df_all = pd.DataFrame()
df_all = pd.concat([df_a, df_b], ignore_index=True)
df_all
```

Out[134]:

		전형	모집인원	지원인원	경쟁률	대학
0		일반고	488	5136	10.52 : 1	부천대학교
1		특성화고	127	1147	9.03 : 1	부천대학교
2		특기자	27	248	9.19 : 1	부천대학교
3		수급자 및 차상위	19	174	9.16 : 1	부천대학교
4		농어촌	9	37	4.11 : 1	부천대학교
5		서해5도	16	0	0.00 : 1	부천대학교
6		북한이탈주민	17	1	0.06 : 1	부천대학교
7		전문대 이상 졸업자	44	172	3.91 : 1	부천대학교
8		총계	747	6915	9.26 : 1	부천대학교
9		일반고 전형	126	1981	15.72 : 1	유한대학교
10		특성화고 전형	105	764	7.28 : 1	유한대학교
11		정원내 소계	231	2745	11.88 : 1	유한대학교
12		전문대학 이상 졸업자 전형	35	18	0.51 : 1	유한대학교
13		기초생활수급자 및 차상위계층 전형	36	154	4.28 : 1	유한대학교
14		서해5도 교육과정 이수자 전형	7	0	0.00 : 1	유한대학교
15		외국인 및 재외국민 전형	36	2	0.06 : 1	유한대학교
16	재직자(특성화고 등을 졸업한 재직자)	재직자(특성화고 등을 졸업한 재직자) 전형	63	7	0.11 : 1	유한대학교
17		정원외 소계	177	181	1.02 : 1	유한대학교
18		총계	408	2926	7.17 : 1	유한대학교

경쟁률 데이터 값 보정하기

후반부의 ': 1' 값 제거\ 자료형을 float로 변경

In [135...]

```
#### '경쟁률' 컬럼 값에서 경쟁률 이 외는 제거
df_all['경쟁률'] = df_all['경쟁률'].apply(lambda x: x.split()[0]) #공백으로 분리 후

#### '경쟁률' 컬럼 자료형을 float로 변경
df_all['경쟁률'] = df_all['경쟁률'].astype(float)
df_all
```

Out[135]:

		전형	모집인원	지원인원	경쟁률	대학
0		일반고	488	5136	10.52	부천대학교
1		특성화고	127	1147	9.03	부천대학교
2		특기자	27	248	9.19	부천대학교
3		수급자 및 차상위	19	174	9.16	부천대학교
4		농어촌	9	37	4.11	부천대학교
5		서해5도	16	0	0.00	부천대학교
6		북한이탈주민	17	1	0.06	부천대학교
7		전문대 이상 졸업자	44	172	3.91	부천대학교
8		총계	747	6915	9.26	부천대학교
9		일반고 전형	126	1981	15.72	유한대학교
10		특성화고 전형	105	764	7.28	유한대학교
11		정원내 소계	231	2745	11.88	유한대학교
12		전문대학 이상 졸업자 전형	35	18	0.51	유한대학교
13		기초생활수급자 및 차상위계층 전형	36	154	4.28	유한대학교
14		서해5도 교육과정 이수자 전형	7	0	0.00	유한대학교
15		외국인 및 재외국민 전형	36	2	0.06	유한대학교
16		재직자(특성화고 등을 졸업한 재직자) 전형	63	7	0.11	유한대학교
17		정원외 소계	177	181	1.02	유한대학교
18		총계	408	2926	7.17	유한대학교

불필요한 행 제거

'전형' 컬럼에 '총계'나 '소계' 값이 들어있는 행은 삭제\ 문자열에서 찾는 문자열의 시작 위치 값 찾기 : `str.find()`

In [22]:

```
## [연습] 문자열에서 찾는 문자열의 시작 위치 값 찾기 : str.find()
str = '계\n소 계\n총 계\n'
str.find('소 계')
```

Out[22]:

2

In [136...]

```
## 불필요한 행 제거 : '전형' 컬럼에 '총계', '소계' 문자열이 포함된 행 제거
df_all = df_all[df_all['전형'].str.find('총계') < 0]
df_all = df_all[df_all['전형'].str.find('소계') < 0]
df_all
```

Out[136]:

		전형	모집인원	지원인원	경쟁률	대학
0		일반고	488	5136	10.52	부천대학교
1		특성화고	127	1147	9.03	부천대학교
2		특기자	27	248	9.19	부천대학교
3		수급자 및 차상위	19	174	9.16	부천대학교
4		농어촌	9	37	4.11	부천대학교
5		서해5도	16	0	0.00	부천대학교
6		북한이탈주민	17	1	0.06	부천대학교
7		전문대 이상 졸업자	44	172	3.91	부천대학교
9		일반고 전형	126	1981	15.72	유한대학교
10		특성화고 전형	105	764	7.28	유한대학교
12		전문대학 이상 졸업자 전형	35	18	0.51	유한대학교
13		기초생활수급자 및 차상위계층 전형	36	154	4.28	유한대학교
14		서해5도 교육과정 이수자 전형	7	0	0.00	유한대학교
15		외국인 및 재외국민 전형	36	2	0.06	유한대학교
16	재직자(특성화고 등을 졸업한 재직자) 전형		63	7	0.11	유한대학교

컬럼 순서 조정

[‘대학’, ‘전형’, ‘모집인원’, ‘지원인원’, ‘경쟁률’] 순으로 조정

In [137...]

```
## '대학' 컬럼이 맨 앞으로 오게 순서 조정
cols = ['대학', '전형', '모집인원', '지원인원', '경쟁률']
df_all = df_all[cols]
df_all
```

Out[137]:

	대학	전형	모집인원	지원인원	경쟁률
0	부천대학교	일반고	488	5136	10.52
1	부천대학교	특성화고	127	1147	9.03
2	부천대학교	특기자	27	248	9.19
3	부천대학교	수급자 및 차상위	19	174	9.16
4	부천대학교	농어촌	9	37	4.11
5	부천대학교	서해5도	16	0	0.00
6	부천대학교	북한이탈주민	17	1	0.06
7	부천대학교	전문대 이상 졸업자	44	172	3.91
9	유한대학교	일반고 전형	126	1981	15.72
10	유한대학교	특성화고 전형	105	764	7.28
12	유한대학교	전문대학 이상 졸업자 전형	35	18	0.51
13	유한대학교	기초생활수급자 및 차상위계층 전형	36	154	4.28
14	유한대학교	서해5도 교육과정 이수자 전형	7	0	0.00
15	유한대학교	외국인 및 재외국민 전형	36	2	0.06
16	유한대학교	재직자(특성화고 등을 졸업한 재직자) 전형	63	7	0.11

3. 대학별 합계 경쟁률 데이터 구축

대학별 모집인원, 지원인원, 총지원인원, 경쟁률 컬럼으로 구성\ 경쟁률이 높은 순으로 보기

In [138...]

##대학별 총모집인원, 총지원인원, 경쟁률 데이터 구축

Out[138]:

대학	총모집인원	총지원인원	경쟁률
부천대학교	747	6915	9.257028
유한대학교	408	2926	7.171569

4-1. [분석] 대학별 합계 경쟁률 비교 그래프 그리기

대학별 합계 경쟁률 순위 막대 그래프 그리기\ 경쟁률이 높은 순으로 그리기

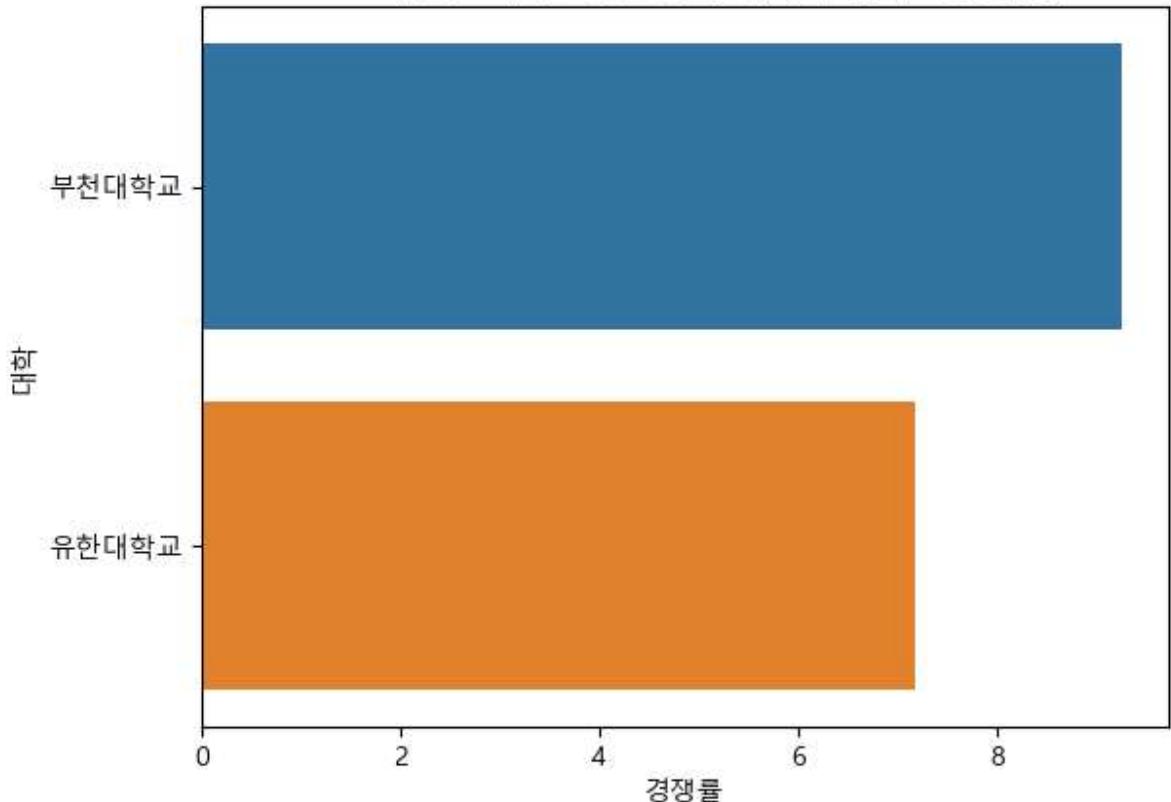
In [140...]

데이터프레임 dml index를 컬럼으로 전환

데이터프레임으로 막대 그래프 그리기

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead  
... if pd.api.types.is_categorical_dtype(vector):  
Out[140]: <Axes: title={'center': '2024학년도 수시2차 입시결과 (대학 전체 : 경쟁률 )'}, xlabel  
='경쟁률', ylabel='대학'>
```

2024학년도 수시2차 입시결과 (대학 전체 : 경쟁률)



4-2. [분석] 대학별 총 지원자 수 비교 그래프 그리기

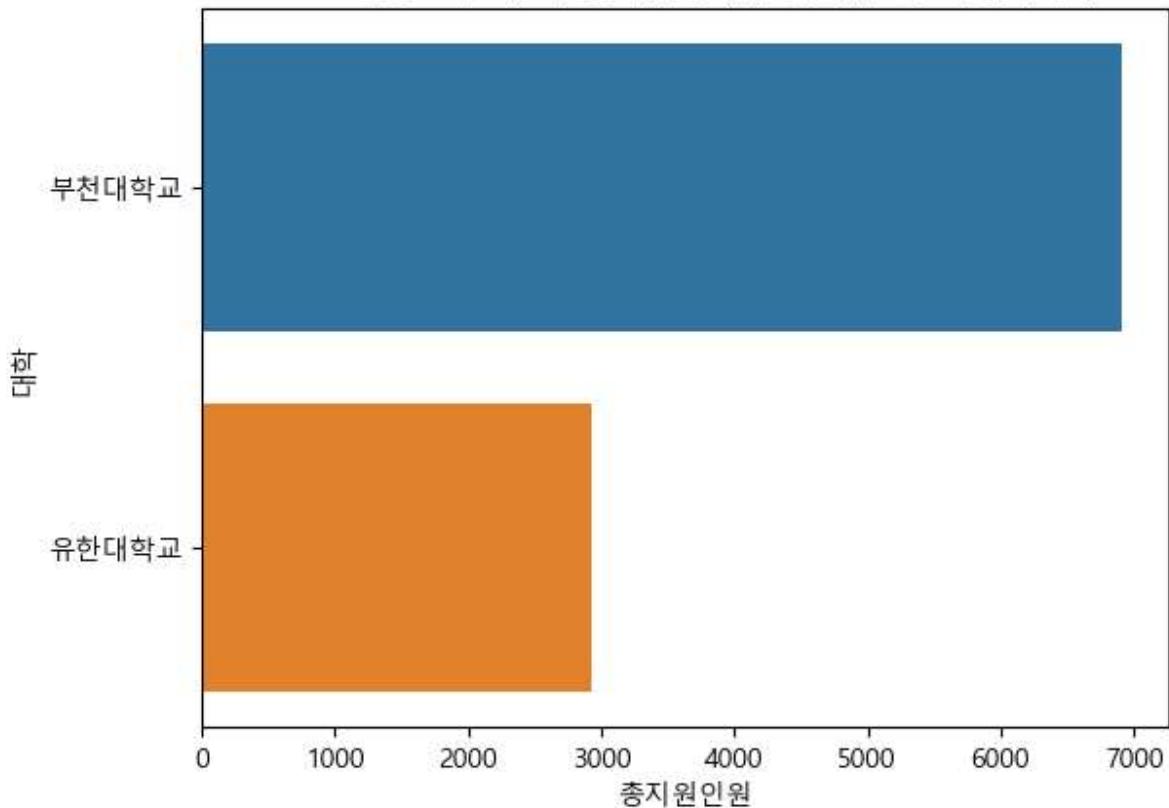
대학별 합계 총 지원자 수 순위 막대 그래프 그리기\총지원인원이 많은 순으로
그리기

```
In [141...]: ## 데이터프레임 df['index']를 컬럼으로 전환
```

```
## 데이터프레임으로 막대 그래프 그리기
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead  
.. if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead  
.. if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead  
.. if pd.api.types.is_categorical_dtype(vector):  
Out[141]: <Axes: title={'center': '2024학년도 수시2차 입시결과 (대학 전체 : 총지원자 수 )'}, x  
label='총지원인원', ylabel='대학'>
```

2024학년도 수시2차 입시결과 (대학 전체 : 총지원자 수)



In []:

[실습-2] Upgrade 하기

1. 기타 대학도 포함시키기 > *Upgrade 대상*

부천대학교, 유한대학교, 연성대학교, 동서울대학교, 동양미래대학교

2. 페이지 정보 데이터 프레임과 반복문을 활용하여 스크랩핑 > *Upgrade 대상*

- > 페이지 정보 데이터 프레임 : 대학명, 페이지 url, 추출할 \index로 구성
- > 반복문 : 스크랩핑할 페이지 정보를 데이터 프레임으로부터 얻어 데이터 추출

html에서 table[] 추출, 불필요한 컬럼 제거, 컬럼명 통일, '대학' 컬럼 추가, 행 병합 순으로 진행

3. 데이터 조정 작업 > [실습-1]과 동일

> 경쟁률 데이터 값 보정하기 : 후반부의 '`:1`' 값 제거, 자료형을 `float`로 변경

> 불필요한 행 제거 : '전형' 컬럼에 '총계'나 '소계' 값이 들어있는 행은 삭제

> 컬럼 순서 조정 : ['대학', '전형', '모집인원', '지원인원', '경쟁률'] 순으로

4. [분석] 대학별 합계 합계 경쟁률 데이터 구축 > [실습-1]과 동일
대학별 모집인원, 지원인원, 총지원인원, 경쟁률 컬럼으로 구성

5. [분석] 대학별 합계 합계 경쟁률 비교 그래프 그리기 > [실습-1]
과 동일

대학별 합계 경쟁률 순위 막대 그래프 그리기\ 대학별 합계 지원자 수 순위 막대 그래프 그리기

In []:

1. 기타 대학도 포함시키기 : 부천대학교, 유한대학교, 연성대학교, 동서울대학교, 동양미래대학교

2. 페이지 정보 데이터 프레임과 반복문을 활용하여 스크랩핑

In [142...]

페이지 정보 데이터 프레임 생성

header 정보 초기화

```
## 페이지 정보 데이터 프레임을 활용해 각 대학의 지원  
현황 데이터를 추출하여 가공 후 새 데이터 프레임에 통합  
df = pd.DataFrame()      # 각 대학의 추출된 데이터 프  
레임 생성  
df_all = pd.DataFrame()  # 하나로 통합할 데이터 프레임  
생성
```

```
for index, row in df_url.iterrows():  
    ## 대학별 데이터 추출 페이지 url 정보 읽어오기
```

지원 현황 데이터 추출

```
## 데이터 보정 작업 : 불필요한 컬럼 제거, 컬럼명  
통일, '대학' 컬럼 추가
```

데이터 통합

```
## 통합된 데이터 확인  
df_all
```

Out[142]:

		전형	모집인원	지원인원	경쟁률	대학
0	일반고 전형		537	2978	5.55 : 1	동양미래대학교
1	특성화고 전형		117	828	7.08 : 1	동양미래대학교
2	전문대학이상 졸업자 전형		132	12	0.09 : 1	동양미래대학교
3	총계		786	3818	4.86 : 1	동양미래대학교
0	일반고 전형		126	1981	15.72 : 1	유한대학교
1	특성화고 전형		105	764	7.28 : 1	유한대학교
2	정원내 소계		231	2745	11.88 : 1	유한대학교
3	전문대학 이상 졸업자 전형		35	18	0.51 : 1	유한대학교
4	기초생활수급자 및 차상위계층 전형		36	154	4.28 : 1	유한대학교
5	서해5도 교육과정 이수자 전형		7	0	0.00 : 1	유한대학교
6	외국인 및 재외국민 전형		36	2	0.06 : 1	유한대학교
7	재직자(특성화고 등을 졸업한 재직자) 전형		63	7	0.11 : 1	유한대학교
8	정원외 소계		177	181	1.02 : 1	유한대학교
9	총계		408	2926	7.17 : 1	유한대학교
0	일반고전형		225	2194	9.75 : 1	동서울대학교
1	특성화고전형		87	691	7.94 : 1	동서울대학교
2	독자기준전형		8	5	0.63 : 1	동서울대학교
3	성인학습자전형		9	15	1.67 : 1	동서울대학교
4	총계		329	2905	8.83 : 1	동서울대학교
0	웹툰만화콘텐츠과(실기)		15	178	11.87 : 1	연성대학교
1	영상콘텐츠과 뉴미디어콘텐츠전공		6	42	7.00 : 1	연성대학교
2	K-POP과		15	60	4.00 : 1	연성대학교
3	사회복지과 아동심리보육전공(면접)		4	10	2.50 : 1	연성대학교
4	사회복지과 아동심리보육전공(비면접)		4	31	7.75 : 1	연성대학교
5	호텔관광과(면접)		10	51	5.10 : 1	연성대학교
6	호텔관광과(비면접)		14	119	8.50 : 1	연성대학교
7	카페·베이커리과		20	87	4.35 : 1	연성대학교
0	일반고		488	5136	10.52 : 1	부천대학교
1	특성화고		127	1147	9.03 : 1	부천대학교

	전형	모집인원	지원인원	경쟁률	대학
2 특기자	27	248	9.19 : 1	부천대학교	
3 수급자 및 차상위	19	174	9.16 : 1	부천대학교	
4 농어촌	9	37	4.11 : 1	부천대학교	
5 서해5도	16	0	0.00 : 1	부천대학교	
6 북한이탈주민	17	1	0.06 : 1	부천대학교	
7 전문대 이상 졸업자	44	172	3.91 : 1	부천대학교	
8 총계	747	6915	9.26 : 1	부천대학교	

```
In [42]: ## [연습] 데이터 프레임의 모양 알아보기 : shape()
df_all.shape #행과 열의 수를 반환
```

```
Out[42]: (30, 5)
```

경쟁률 데이터 값 보정하기

후반부의 ': 1' 값 제거\ 자료형을 float로 변경

불필요한 행 제거

'전형' 컬럼에 '총계'나 '소계' 값이 들어있는 행은 삭제

```
In [143... ##### '경쟁률' 컬럼 자료형을 float로 변경
```

```
## 불필요한 행 제거 : '전형' 컬럼에 '총계'나 '소계' 문자열이 포함된 행 제거
```

```
df_all
```

Out[143]:

	전형	모집인원	지원인원	경쟁률	대학
0	일반고 전형	537	2978	5.55	동양미래대학교
1	특성화고 전형	117	828	7.08	동양미래대학교
2	전문대학이상 졸업자 전형	132	12	0.09	동양미래대학교
0	일반고 전형	126	1981	15.72	유한대학교
1	특성화고 전형	105	764	7.28	유한대학교
3	전문대학 이상 졸업자 전형	35	18	0.51	유한대학교
4	기초생활수급자 및 차상위계층 전형	36	154	4.28	유한대학교
5	서해5도 교육과정 이수자 전형	7	0	0.00	유한대학교
6	외국인 및 재외국민 전형	36	2	0.06	유한대학교
7	재직자(특성화고 등을 졸업한 재직자) 전형	63	7	0.11	유한대학교
0	일반고전형	225	2194	9.75	동서울대학교
1	특성화고전형	87	691	7.94	동서울대학교
2	독자기준전형	8	5	0.63	동서울대학교
3	성인학습자전형	9	15	1.67	동서울대학교
0	웹툰만화콘텐츠과(실기)	15	178	11.87	연성대학교
1	영상콘텐츠과 뉴미디어콘텐츠전공	6	42	7.00	연성대학교
2	K-POP과	15	60	4.00	연성대학교
3	사회복지과 아동심리보육전공(면접)	4	10	2.50	연성대학교
4	사회복지과 아동심리보육전공(비면접)	4	31	7.75	연성대학교
5	호텔관광과(면접)	10	51	5.10	연성대학교
6	호텔관광과(비면접)	14	119	8.50	연성대학교
7	카페·베이커리과	20	87	4.35	연성대학교
0	일반고	488	5136	10.52	부천대학교
1	특성화고	127	1147	9.03	부천대학교
2	특기자	27	248	9.19	부천대학교
3	수급자 및 차상위	19	174	9.16	부천대학교
4	농어촌	9	37	4.11	부천대학교
5	서해5도	16	0	0.00	부천대학교
6	북한이탈주민	17	1	0.06	부천대학교
7	전문대 이상 졸업자	44	172	3.91	부천대학교

대학별 합계 합계 경쟁률 데이터 구축

대학별 모집인원, 지원인원, 총지원인원, 경쟁률 컬럼으로 구성\ 경쟁률이 높은 순으로 보기

In [144...]

대학별 총모집인원, 총지원인원, 경쟁률 데이터 구축

Out[144]:

총모집인원 총지원인원 경쟁률

대학	총모집인원	총지원인원	경쟁률
부천대학교	747	6915	9.257028
동서울대학교	329	2905	8.829787
유한대학교	408	2926	7.171569
연성대학교	88	578	6.568182
동양미래대학교	786	3818	4.857506

[분석] 대학별 합계 합계 경쟁률 비교 그래프 그리기

경쟁율 내림차 순으로 그리기

In [145...]

```
## 데이터프레임을 정렬하고, index를 컬럼으로 전환
```

```
## 데이터프레임으로 막대 그래프 그리기
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead
```

```
... if pd.api.types.is_categorical_dtype(vector):
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead
```

```
... if pd.api.types.is_categorical_dtype(vector):
```

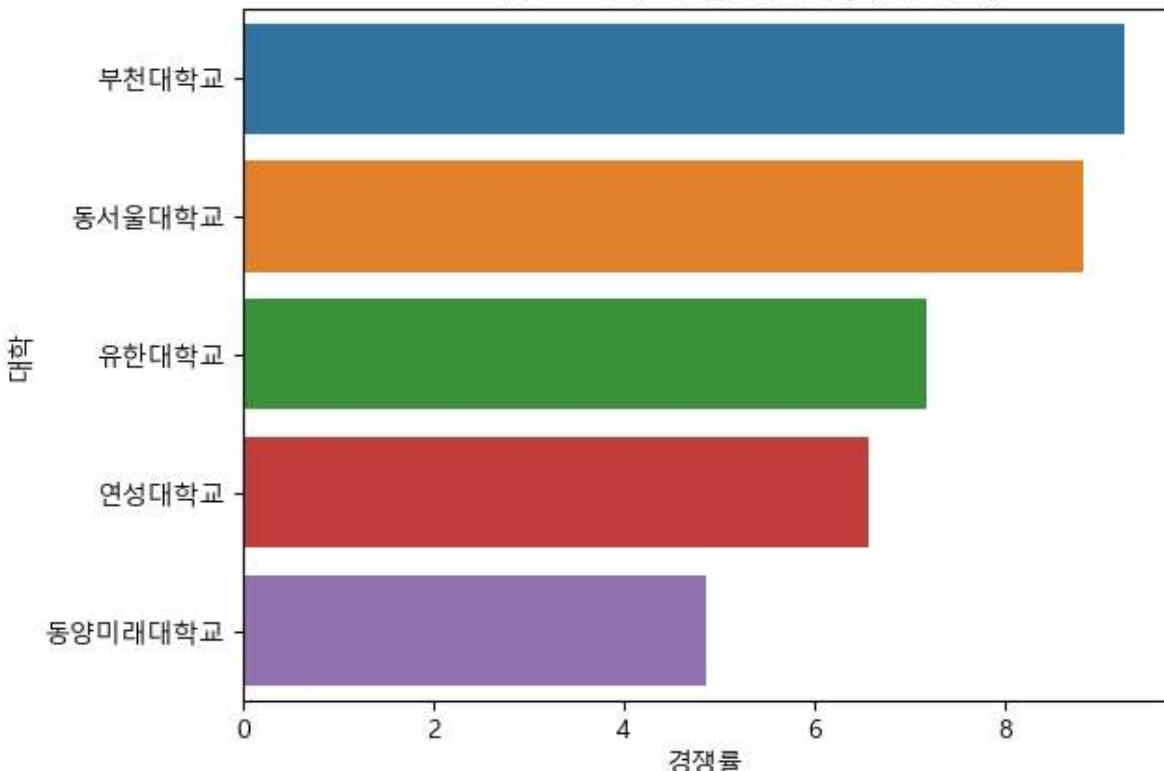
```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead
```

```
... if pd.api.types.is_categorical_dtype(vector):
```

Out[145]:

```
<Axes: title={'center': '2024학년도 수시2차 입시결과 (대학 전체)'}, xlabel='경쟁률',  
ylabel='대학'>
```

2024학년도 수시2차 입시결과 (대학 전체)



[분석] 대학별 합계 지원자 수 비교 그래프 그리기

대학별 합계 총 지원자 수 순으로 그리기

In [146...]

```
## 데이터프레임을 정렬하고, index를 컬럼으로 전환
```

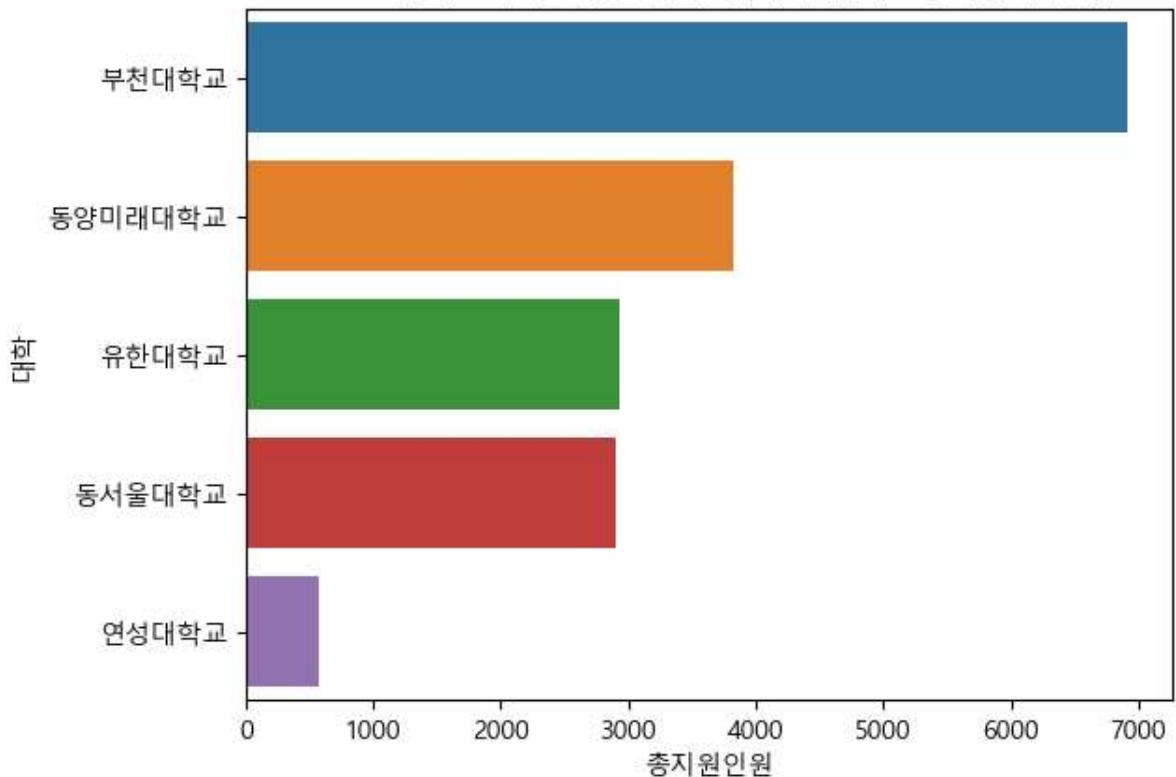
```
## 데이터프레임으로 막대 그래프 그리기
```

```
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):  
C:\Users\ADMIN\anaconda3\lib\site-packages\seaborn\oldcore.py:1498: FutureWarning:  
is_categorical_dtype is deprecated and will be removed in a future version. Use isin  
stance(dtype, CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):
```

Out[146]:

```
<Axes: title={'center': '2024학년도 수시2차 입시결과 (대학 전체 : 총지원자 수 )'}, x  
label='총지원인원', ylabel='대학'>
```

2024학년도 수시2차 입시결과 (대학 전체 : 총지원자 수)



In []: