[데이터 분석 및 시각화 활용 기초 강의 포트폴리오]

- 고등학교 3학년 9월 모의평가 성적 분석 자료 만들기-

- 장 인 영-

주제 선정 이유

- 현재 고등학교 3학년 교사
- 현업에 적용할 수 있는 부분을 시도
- 9월 모의고사 결과를 분석
- 앞으로의 학습 지도 방향 모색

Code Review

In [6]:
```
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.rc('font', family='Malgun Gothic')
```

In [7]:
```
df= pd.read_csv('./9월 모의고사.csv', index_col=0, encoding='euc-kr')
```

In [8]:
```
df.head()
```

Out[8]:

학번	성명	국어선택	국어표준점수	국어백분위	국어등급	수학선택	수학표준점수	수학백분위	수학등급	영어등급	한국사등급	탐구1선택	탐구1표준점수	탐구1백분위	탐구1등급	탐구2선택	탐구2표준점수	탐구2백분위	탐구2등급
3101	***	언어와 매체	95	34	6	확률과 통계	97	43	5	3	7	생활과윤리	38	17	7	사회·문화	52	53	5
3102	***	화법과 작문	95	34	6	확률과 통계	92	36	6	4	1	생활과윤리	42	25	6	사회·문화	38	17	7
3103	***	화법과 작문	96	36	6	확률과 통계	75	16	7	6	1	동아시아사	52	56	5	세계사	49	54	5

Code Review

1. 과목 선택 비율

In [9]:
```
# 국어 선택과목 별로 카운트
k_choice = pd.DataFrame(df['국어선택'].value_counts(ascending=True))
```

In [10]:
```
k_choice
```

Out[10]:

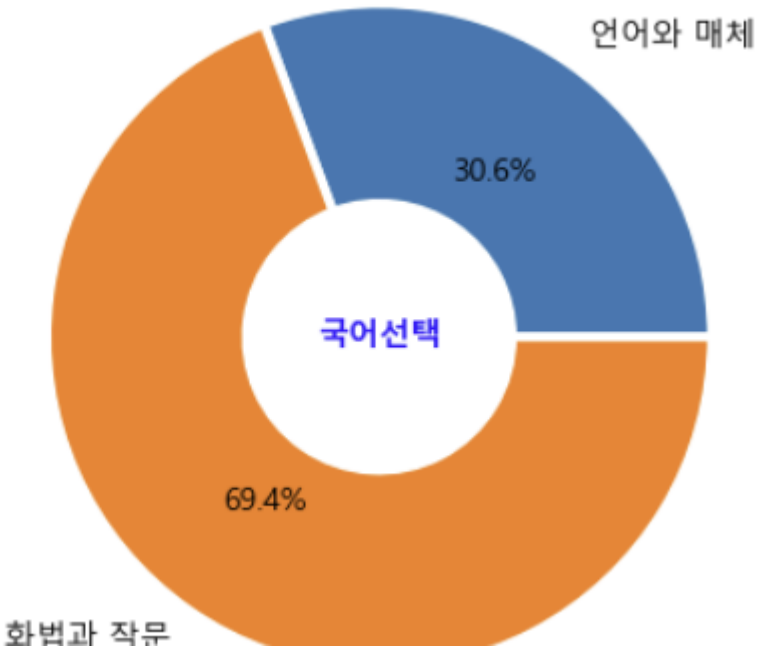
국어선택	count
언어와 매체	38
화법과 작문	86

In [11]:
```
# 차트 영역 지정하기
plt.figure(figsize=(6,5))
```

Out[11]:
```
<Figure size 600x500 with 0 Axes>

<Figure size 600x500 with 0 Axes>
```

In [12]:

```
# 파이 그래프 그리기

wedgeprops = {'linewidth' : 3, 'edgecolor' : 'white', 'width':0.6}

plt.pie(
    x = k_choice['count'],₩
    labels = k_choice.index,₩
    autopct = '%.1f%%',₩
    wedgeprops = wedgeprops
)
plt.text(0,0,'국어선택', ha='center', va='center', fontsize=10, fontweight='bold', color="blue")


# 차트 표시하기
plt.show()
```



Code Review

In [14]:
```
df.isnull().sum()  # null 값 확인 -> null 이 있으면 0으로 대체
```

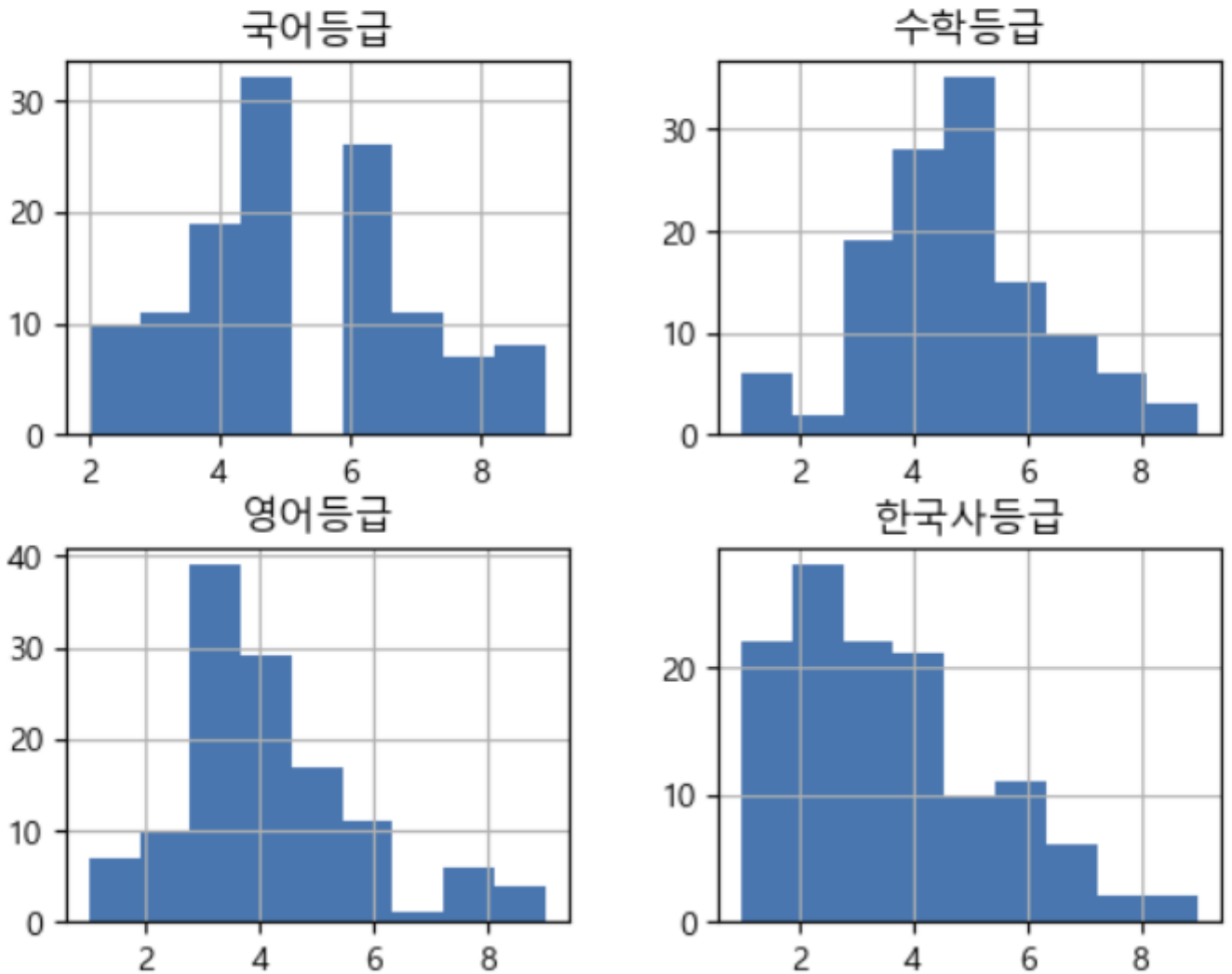
Out[14]:
```
성명          0
국어선택        0
국어표준점수      0
국어백분위       0
국어등급        0
수학선택        0
수학표준점수      0
수학백분위       0
수학등급        0
영어등급        0
한국사등급       0
탐구1선택       0
탐구1표준점수     0
탐구1백분위      0
탐구1등급       0
탐구2선택       0
탐구2표준점수     0
탐구2백분위      0
탐구2등급       0
dtype: int64
```

In [15]:
```
df_class = df[['국어등급','수학등급','영어등급','한국사등급']]
```

Code Re

In [16]:
```
df_class.hist(bins=9)  # 등급은 9등급까지 있으므로 bins = 9
```

Out[16]:
```
array([[<AxesSubplot:title={'center':'국어등급'}>,
        <AxesSubplot:title={'center':'수학등급'}>],
       [<AxesSubplot:title={'center':'영어등급'}>,
        <AxesSubplot:title={'center':'한국사등급'}>]], dtype=object)
```



2. 과목별 등급 분포 그래프

국어 등급 분포

In [17]:
```
# 간단한 막대 그래프
plt.figure(figsize = (6,3))

# 국어 등급 별로 카운트
k_class = pd.DataFrame(df['국어등급'].value_counts(ascending=True))
```

```
<Figure size 600x300 with 0 Axes>
```

In [18]:
```
# 오름차순으로 정렬하기

k_class_sort = pd.DataFrame(k_class.sort_index())
k_class_sort2 = k_class_sort.reset_index()
k_class_sort2
```

Out[18]:

	국어등급	count
0	2	10
1	3	11
2	4	19
3	5	32
4	6	26
5	7	11
6	8	7
7	9	8

Code Review

In [19]:

```
# 1등급이 없으므로 추가해주자. 1등급은 0명이다.

k_class_sort2.loc[8] = [1, 0]
k_class_sort2

k_class_sort2.sort_values(by = '국어등급', ascending = True, inplace = True)
k_class_sort2
```

Out[19]:

	국어등급	count
8	1	0
0	2	10
1	3	11
2	4	19
3	5	32
4	6	26
5	7	11
6	8	7
7	9	8

Code Review

In [20]:
```
k_class_sort3 = k_class_sort2.reset_index()
k_class_sort3
```

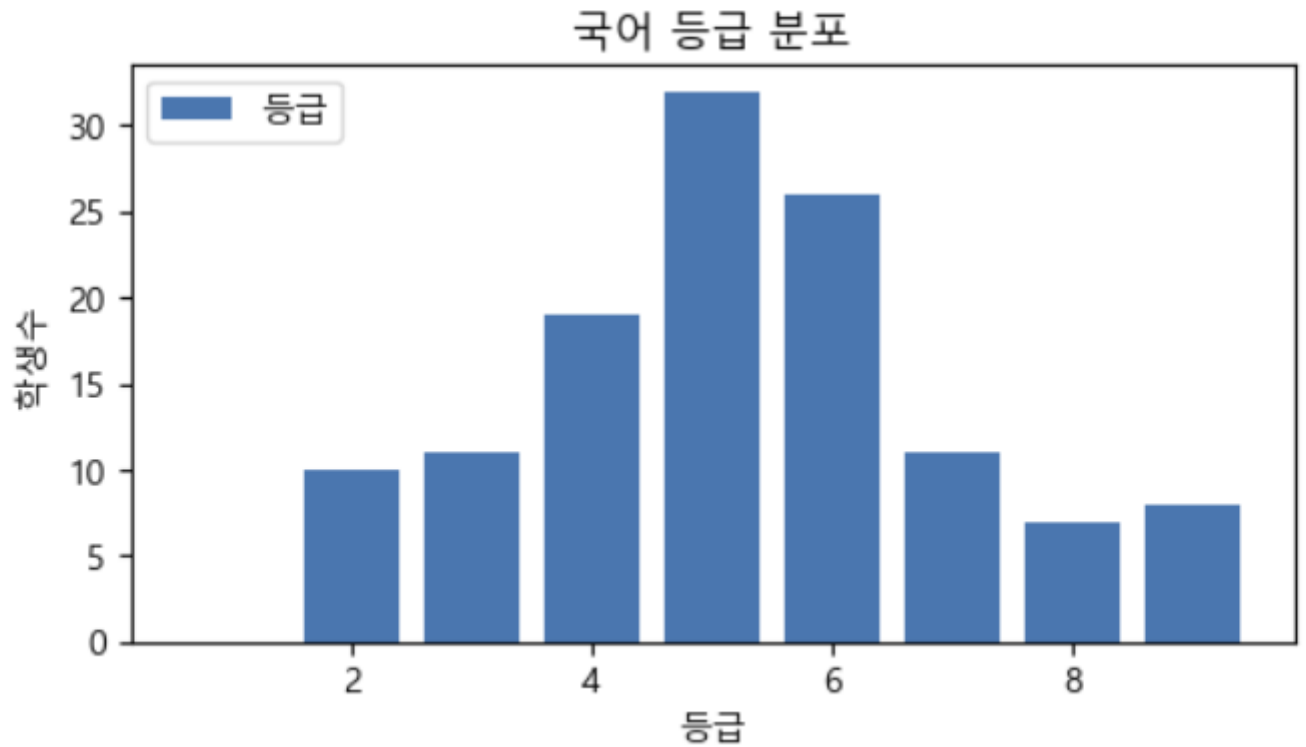
Out[20]:

	index	국어등급	count
0	8	1	0
1	0	2	10
2	1	3	11
3	2	4	19
4	3	5	32
5	4	6	26
6	5	7	11
7	6	8	7
8	7	9	8

Code Review

In [21]:

```
# 막대 그래프 그리기
plt.figure(figsize = (6,3))
plt.bar(k_class_sort3['국어등급'], k_class_sort3['count'], label = '등급')

# 제목, 축 라벨, 범례 추가
plt.title('국어 등급 분포')
plt.xlabel('등급')
plt.ylabel("학생수")
plt.legend(loc="upper left")
plt.show()
```



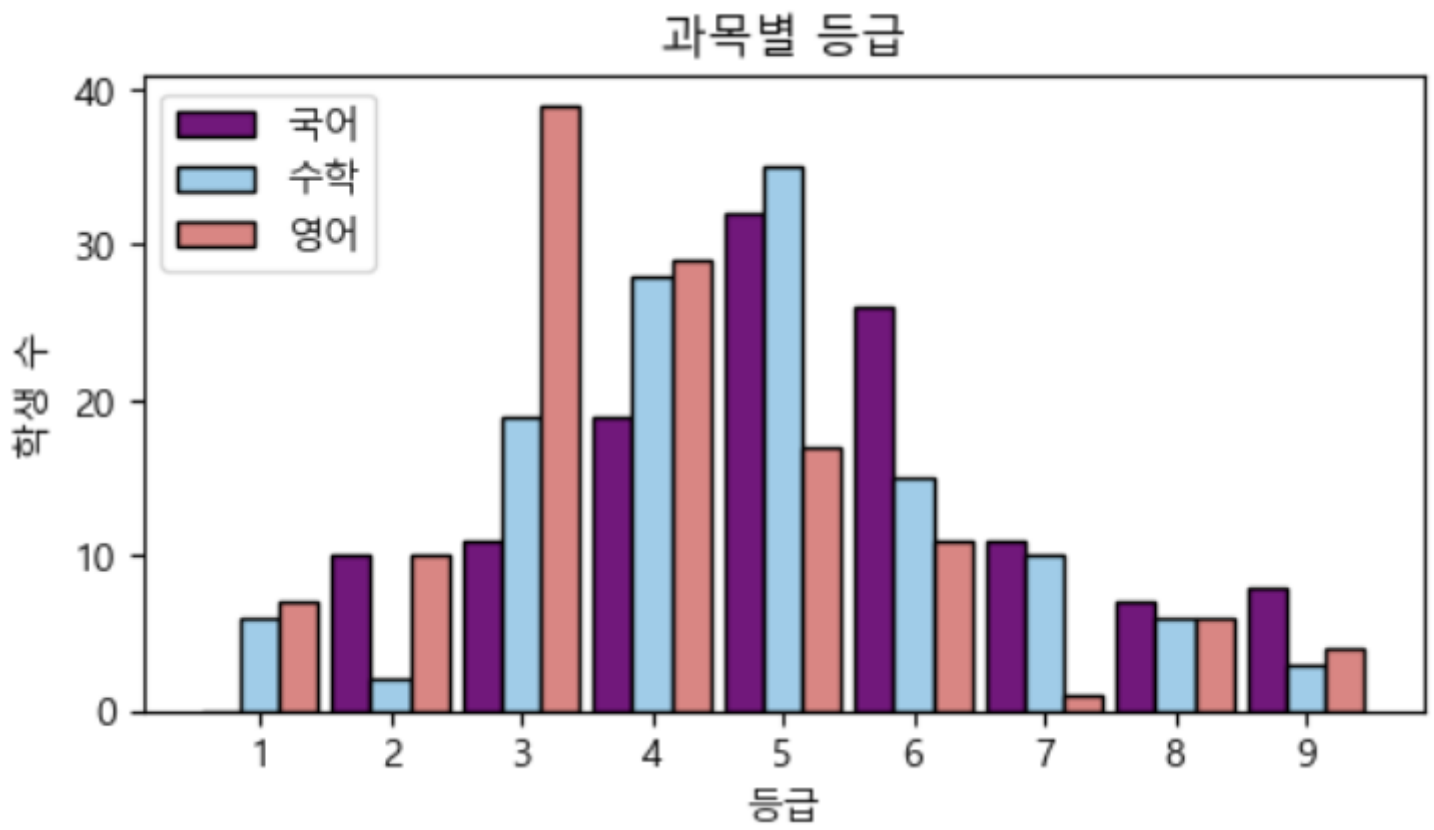
Code Review

국영수 종합하기

In [29]:

```
import numpy as np

# x 위치를 생성하여 막대 간격 조정
x = np.arange(len(e_class_sort2['영어등급']))
width = 0.3# 막대 너비

# 그래프 생성
plt.figure(figsize=(6, 3))

# sales는 x의 위치에서 width 만큼 -위치, expenses는 x의 위치에서 width 만큼 -위치
plt.bar(x - width, k_class_sort3['count'], width=width, color="purple", edgecolor="black", label="국어")
plt.bar(x, m_class_sort2['count'], width=width, color="skyblue", edgecolor="black", label="수학")
plt.bar(x + width, e_class_sort2['count'], width=width, color="lightcoral", edgecolor="black", label="영어")

# x축 레이블 설정
plt.xlabel("등급")
plt.ylabel("학생 수")
plt.title("과목별 등급")
# X축 눈금의 레이블 등급으로 지정
plt.xticks(x, e_class_sort2['영어등급'])

# 범례 추가
plt.legend(loc="upper left")
plt.show()
```

Code Review



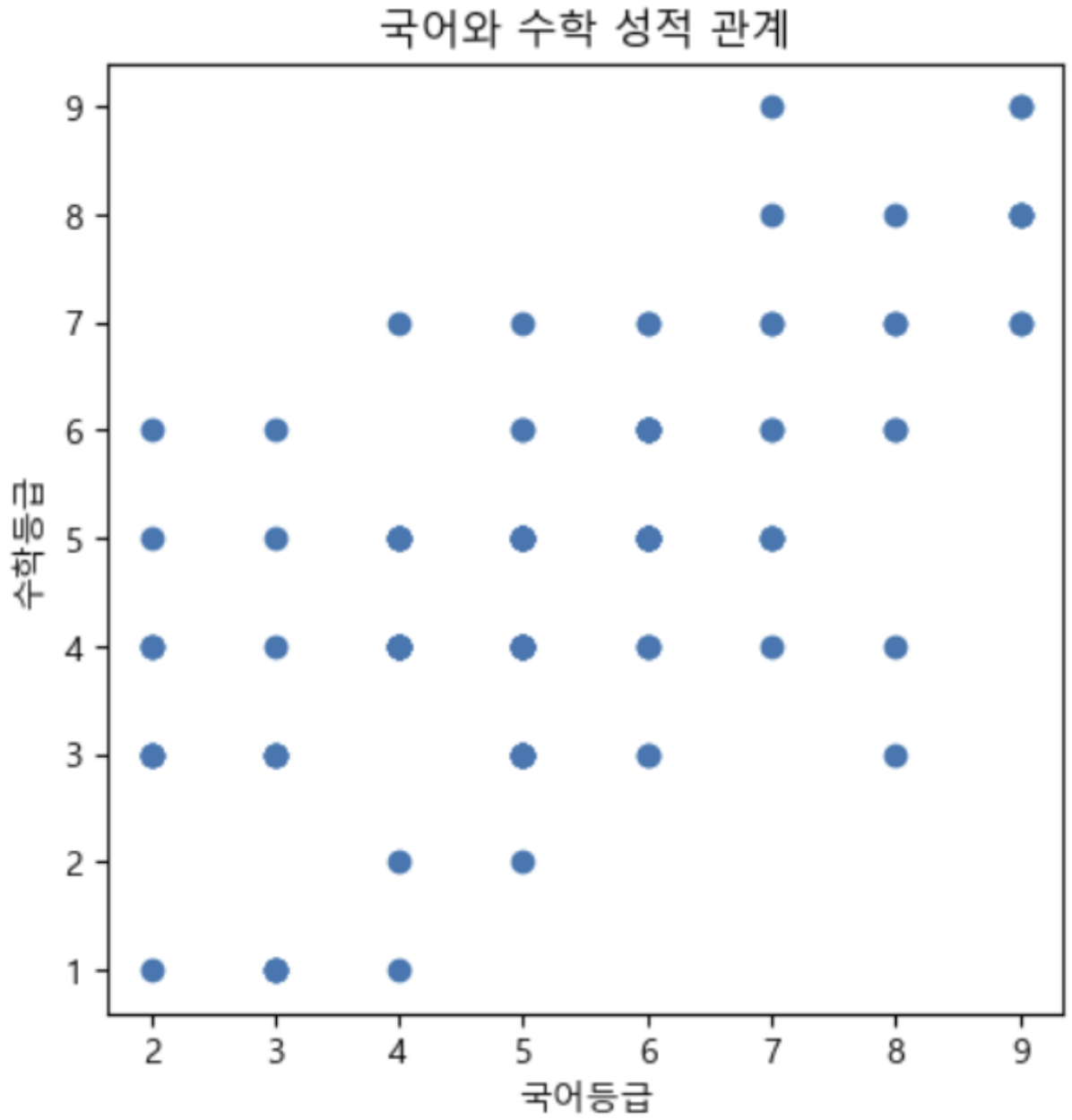
Code Review

3. 국어와 수학 성적 상관관계 분석

In [30]:

```
# 기본 Scatter Plot
plt.figure(figsize=(5, 5))
plt.scatter(df["국어등급"], df["수학등급"])

# 제목과 축 라벨 설정
plt.title("국어와 수학 성적 관계")
plt.xlabel("국어등급")
plt.ylabel("수학등급")
plt.show()


# 중복되는 데이터 어떻게 해결할까? ---> 문제...
```

Code Review



Code Review

4. 표준점수 합계, 정렬

In [31]:
```
df_sc = df[['국어표준점수', '수학표준점수', '탐구1표준점수', '탐구2표준점수']]  # 표준점수만 불러오기
```

In [32]:
```
df_sc.head()
df_sc.sum(axis=0)
df['탐구2표준점수']  ## sum을 하려니까 데이터 타입 오류 뜸
```

Out[32]:
```
학번
3101    52
3102    38
3103    49
3104    37
3105    62
       ..
3617    37
3618    66
3619    63
3620    46
3621    42
Name: 탐구2표준점수, Length: 124, dtype: object
```

Code Review

In [33]:
```
# '탐구1 표준점수', '탐구2 표준점수' 데이터를 object -> int로 바꾸어야 계산 가능

df_sc = df_sc.astype(int)
```

```
    134
    135     return arr.astype(dtype, copy=copy)

ValueError: invalid literal for int() with base 10: '-'
```

Code

In [34]:
```
df_sc = df_sc.replace({'탐구1표준점수': '-'}, 0)   # '-' 때문에 'object'로 인식한 것이었음.
df_sc = df_sc.replace({'탐구2표준점수': '-'}, 0)
df_sc = df_sc.astype(int)
```

In [35]:
```
df_sc.dtypes  #바뀐 데이터 확인
```

Out[35]:
```
국어표준점수     int32
수학표준점수     int32
탐구1표준점수    int32
탐구2표준점수    int32
dtype: object
```

In [36]:
```
df_sc.sum(axis=1)   # 표준점수 합계 잘 나오는지 확인
```

Out[36]:
```
학번
3101    282
3102    267
3103    272
3104    231
3105    356
       ...
3617    294
3618    378
3619    360
3620    310
3621    318
Length: 124, dtype: int64
```

Code Review

In [37]:
```
df['표준점수 합계'] = df_sc.sum(axis=1)
```

In [38]:
```
df.head()
```

Out[38]:

학번	성명	국어 선택	국어표준점수	국어 백분위	국어 등급	수학 선택	수학표준점수	수학 백분위	수학 등급	영어 등급	한국사등급	탐구1 선택	탐구1 표준점수	탐구1 백분위	탐구 1등급	탐구 2선택	탐구2 표준점수	탐구2 백분위	탐구 2등급	표준점수 합계
3101	***	언어와 매체	95	34	6	확률과 통계	97	43	5	3	7	생활과윤리	38	17	7	사회·문화	52	53	5	282
3102	***	화법과 작문	95	34	6	확률과 통계	92	36	6	4	1	생활과윤리	42	25	6	사회·문화	38	17	7	267
3103	***	화법과 작문	96	36	6	확률과 통계	75	16	7	6	1	동아시아사	52	56	5	세계사	49	54	5	272
3104	***	화법과 작문	66	10	8	확률과 통계	90	34	6	5	3	생활과윤리	38	17	7	사회·문화	37	15	7	231
		언어																		

Code Review

```
In [39]: df.sort_values(by = '표준점수 합계', ascending = False, inplace = True)  # 표준점수대로 내림차순 정렬

In [40]: df.head(10)   # 상위권 10명 보기
```

학번																				
3520	***	언어와 매체	125	95	2	미적분	133	98	1	3	4	생활과윤리	64	94	1	사회·문화	65	95	1	387
3511	***	언어와 매체	119	81	3	미적분	133	98	1	2	3	생명과학I	64	94	2	지구과학I	64	92	2	380
3516	***	언어와 매체	120	84	3	미적분	130	95	1	3	4	한국지리	66	97	1	지구과학I	64	92	2	380
3618	***	언어와 매체	124	93	2	미적분	124	86	3	3	1	화학I	64	92	2	생명과학I	66	97	1	378
3603	***	언어와 매체	119	81	3	미적분	135	99	1	1	1	화학I	62	86	3	생명과학I	56	68	4	372
언어																				

Code Review

```
In [41]: df.info()  # 124개의 행 데이터 확인
```

...

```
In [42]: num = list(range(1,125))
         num
```

...

```
In [43]: df.insert(loc = 1, column ='등수_표점', value = num)   # 등수 열 추가
```

```
In [44]: df.head()
```

Out[44]:

학번	성명	등수_표점	국어 선택	국어표준점수	국어 백분위	국어 등급	수학 선택	수학표준점수	수학 백분위	수학 등급	...	한국사등급	탐구1선택	탐구1 표준점수	탐구1 백분위	탐구1 등급	탐구2선택	탐구2 표준점수	탐구2 백분위	탐구2 등급	표준점수 합계
3520	***	1	언어와 매체	125	95	2	미적분	133	98	1	...	4	생활과윤리	64	94	1	사회·문화	65	95	1	387
3511	***	2	언어와 매체	119	81	3	미적분	133	98	1	...	3	생명과학I	64	94	2	지구과학I	64	92	2	380

Code Review

5. 합격 가능성 여부 파악하기

In [45]:
```
# 표준점수 합계가 350 이상이고, 영어등급이 2등급 이하인 학생이 합격 가능성이 있다.

df_filter = df[(df['표준점수 합계'] >= 350) & (df['영어등급'] <= 2) ]
```

In [46]:
```
df_filter
```

Out[46]:

학번	성명	등수_표점	국어 선택	국어표준점수	국어 백분위	국어 등급	수학 선택	수학표준점수	수학 백분위	수학 등급	...	한국사등급	탐구1 선택	탐구1 표준점수	탐구1 백분위	탐구1 등급	탐구2 선택	탐구2 표준점수	탐구2 백분위	탐구2 등급	표준점수 합계
3511	***	2	언어와 매체	119	81	3	미적분	133	98	1	...	3	생명과학I	64	94	2	지구과학I	64	92	2	380
3603	***	5	언어와 매체	119	81	3	미적분	135	99	1	...	1	화학I	62	86	3	생명과학I	56	68	4	372
3108	***	7	화법과 작문	113	68	4	미적분	127	91	2	...	1	경제	64	88	3	사회·문화	60	81	3	364
3203	***	8	화법과 작문	124	93	2	미적분	114	70	4	...	1	생활과윤리	64	94	1	세계사	61	80	3	363

Code Review

학번 입력했을 때, 합격 가능 불가능 여부 파악하기

In [47]:
```
df_last = df.reset_index()
df_last['학번']
```

Out[47]:
```
0      3520
1      3511
2      3516
3      3618
4      3603
       ...
119    3209
120    3314
121    3419
122    3119
123    3215
Name: 학번, Length: 124, dtype: int64
```

In [48]:
```
student_num = input('학번을 입력하세요. ex)3101')

df_last[df_last['학번'] == int(student_num)]  # 정수형으로 바꾸어주어야 함
```

Code Review

In [53]:
```
student_num = input('학번을 입력하세요. ex)3101')

df_last[df_last['학번'] == int(student_num)]  # 정수형으로 바꾸어주어야 함
```

학번을 입력하세요. ex)31013403

Out[53]:

	학번	성명	등수_표점	국어 선택	국어표준점수	국어 백분위	국어 등급	수학 선택	수학표준점수	수학 백분위	...	한국사등급	탐구1 선택	탐구1 표준점수	탐구1 백분위	탐구 1등급	탐구 2선택	탐구2 표준점수	탐구2 백분위	탐구 2등급	표준점수 합계
19	3403	***	20	화법과작문	124	93	2	확률과통계	104	54	...	5	생활과윤리	58	73	4	정치와법	63	88	2	349

1 rows × 22 columns

In [*]:
```
student_num = input('학번을 입력하세요. ex)3101')

student_info = df_last[df_last['학번'] == int(student_num)]  # 정수형으로 바꾸어주어야 함

student_info
```

학번을 입력하세요. ex)3101 3403

In [50]:
```
print(student_info['표준점수 합계'])
print(student_info['영어등급'])
```

In [51]:
```
condition1 = student_info['표준점수 합계'] >= 350
condition2 = student_info['영어등급'] <= 2

print(condition1)
print(condition2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~₩AppData₩Local₩Temp₩ipykernel_1640₩2850864466.py in <module>
----> 1 condition1 = student_info['표준점수 합계'] >= 350
      2 condition2 = student_info['영어등급'] <= 2
      3
      4 print(condition1)
      5 print(condition2)

NameError: name 'student_info' is not defined
```

In [52]:
```
if condition1 & condition2:
    print('합격입니다.')
else:
    print('불합격입니다.')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~₩AppData₩Local₩Temp₩ipykernel_1640₩1956181814.py in <module>
----> 1 if condition1 & condition2:
      2     print('합격입니다.')
```

한계점 및 해보고 싶은 점

- 마지막 dataframe 에서 if문을 사용하는 것에서 오류를 해결하지 못함.
- 탐구1, 탐구2 데이터를 병합하여 통계를 내는 것을 해보고 싶음.
- 최저학력 기준 등급을 맞추는 프로그래밍을 해보고 싶음.