

```
In [1]: import pandas as pd
import numpy as np

# 샘플 데이터 생성
np.random.seed(0)

hanwha_players_2025 = [
    '류현진', '문동주', '김서현', '주현상', '엄상백', '김승일', '한승혁',
    '이상규', '조동욱', '윤대경', '이민우', '장시환', '황준서', '폰세',
    '장민재', '김종수', '장지수', '정우주', '박상원', '배동현', '성지훈',
    '이태양', '김민우', '김기중', '권민규', '김규연', '김도빈', '김범수',
    '배민서', '와이스',
    '안치홍', '노시환', '김인환', '문현빈', '심우준', '정민규', '하주석',
    '이도윤', '채은성', '황영록', '배승수', '김태연', '이진영', '이상혁',
    '유로결', '최인호', '이원석', '임종찬', '최준서', '리베라토'
]

subjects = ['수학', '영어', '과학', '국어']

data = {
    '이름': np.repeat(hanwha_players_2025, 4), # 각 선수를 4번씩 반복
    '과목': np.tile(subjects, 50), # 4개 과목을 50번 반복
    '점수': np.random.randint(60, 101, 200), # 200개의 점수 데이터
    '학년': np.random.randint(1, 4, 200) # 1-3학년
}

df = pd.DataFrame(data)
```

In [2]: df

	이름	과목	점수	학년
0	류현진	수학	60	2
1	류현진	영어	63	2
2	류현진	과학	63	3
3	류현진	국어	99	1
4	문동주	수학	69	2
...
195	최준서	국어	70	2
196	리베라토	수학	97	2
197	리베라토	영어	88	3
198	리베라토	과학	100	3
199	리베라토	국어	62	2

200 rows × 4 columns

groupby를 사용하여 각 학생의 평균 점수를 계산하고, 평균 점수가 가장 높은 상위 3명의 학생을 골라 출력해주세요.

```
In [14]: mean = df.groupby('이름')['점수'].mean().reset_index()
top3 = mean.sort_values('점수', ascending=False).head(3)
print(top3)
```

	이름	점수
19	배승수	91.00
9	김종수	90.75
45	폰세	87.50

groupby와 apply를 사용하여 각 과목별로 학년 간 평균 점수 차이가 가장 큰 과목을 구해보세요.

```
In [13]: result = (
    df.groupby('과목')
        .apply(lambda x: x.groupby('학년')['점수'].mean().max() -
               x.groupby('학년')['점수'].mean().min(),
              include_groups=False)
        .reset_index(name='학년간_평균점수차')
)

# 점수 차이가 가장 큰 과목 추출
top_subject = result.loc[result['학년간_평균점수차'].idxmax(), '과목']

print("각 과목별로 학년 간 점수 차이가 가장 큰 과목:")
print(top_subject)
```

각 과목별로 학년 간 점수 차이가 가장 큰 과목:
과학

apply를 사용하여 각 학생의 점수에 대해 다음 기준으로 등급을 매기는 새로운 열을 만들어 봅시다.

- 95점 이상: 'A+'
- 90-94점: 'A'
- 85-89점: 'B+'
- 80-84점: 'B'
- 75-79점: 'C+'
- 70-74점: 'C'
- 70점 미만: 'F'

그리고 각 등급별 학생 수를 계산하세요

```
In [9]: def grade(score):
    if score >= 95:
        return 'A+'
    elif score >= 90:
        return 'A'
    elif score >= 85:
        return 'B+'
    elif score >= 80:
        return 'B'
    elif score >= 75:
        return 'C+'
    elif score >= 70:
        return 'C'
```

```

else:
    return 'F'

# apply로 등급 컬럼 추가
df['등급'] = df['점수'].apply(grade)

# 등급별 학생 수 계산
grade_order = ['A+', 'A', 'B+', 'B', 'C+', 'C', 'F']
grade_counts = df['등급'].value_counts().reindex(grade_order, fill_value=0

print("등급별 학생 수:")
print(grade_counts)

```

등급별 학생 수:

등급	학생 수
A+	37
A	22
B+	11
B	28
C+	24
C	28
F	50

Name: count, dtype: int64

groupby와 apply를 사용하여 각 과목별로 상위 20% 학생의 점수 평균을 계산하세요.

```

In [10]: def top20_avg(group):
    # 점수 내림차순 정렬
    group = group.sort_values('점수', ascending=False)
    # 상위 20% 인원 수 계산
    top20 = max(1, int(len(group) * 0.2)) # 최소 1명 보장
    # 상위 20% 학생 점수 평균 반환
    return group.head(top20)['점수'].mean()

    # 과목별로 상위 20% 평균 계산
top20_means = df.groupby('과목').apply(top20_avg, include_groups=False).re

print("각 과목별 상위 20% 학생의 평균 점수:")
print(top20_means)

```

각 과목별 상위 20% 학생의 평균 점수:

과목	상위20%_평균점수
0 과학	98.2
1 국어	95.8
2 수학	97.1
3 영어	96.8