

## [과제] Ch06 데이터 전처리

```
In [1]: ## 실습용 데이터 프레임 생성
import pandas as pd
exam = pd.read_csv('exam_csv.csv')
exam.head()
```

```
Out[1]:
```

	id	nclass	math	english	science
0	1	1	50	98	50
1	2	1	60	97	60
2	3	1	45	86	78
3	4	1	30	98	58
4	5	2	25	80	65

**[실습 6-1] exam 데이터 프레임에 'total'과 'mean', 'result' 파생변수가 추가된 new\_exam 데이터 프레임 생성**

> 'total'은 과목의 합계

> 'mean'은 전체 과목의 평균

> 'result'는 'mean' >= 60 이면 'pass', 아니면 'fail'

```
In [17]: ### [실습 6-1] exam 데이터 프레임에 'total'과 'mean', 'result' 파생변수가 추가
```

```
new_exam
```

Out[17]:

	id	nclass	math	english	science	total	mean	result
<b>0</b>	1	1	50	98	50	198	66.000000	pass
<b>1</b>	2	1	60	97	60	217	72.333333	pass
<b>2</b>	3	1	45	86	78	209	69.666667	pass
<b>3</b>	4	1	30	98	58	186	62.000000	pass
<b>4</b>	5	2	25	80	65	170	56.666667	fall
<b>5</b>	6	2	50	89	98	237	79.000000	pass
<b>6</b>	7	2	80	90	45	215	71.666667	pass
<b>7</b>	8	2	90	78	25	193	64.333333	pass
<b>8</b>	9	3	20	98	15	133	44.333333	fall
<b>9</b>	10	3	50	98	45	193	64.333333	pass
<b>10</b>	11	3	65	65	65	195	65.000000	pass
<b>11</b>	12	3	45	85	32	162	54.000000	fall
<b>12</b>	13	4	46	98	65	209	69.666667	pass
<b>13</b>	14	4	48	87	12	147	49.000000	fall
<b>14</b>	15	4	75	56	78	209	69.666667	pass
<b>15</b>	16	4	58	98	65	221	73.666667	pass
<b>16</b>	17	5	65	68	98	231	77.000000	pass
<b>17</b>	18	5	80	78	90	248	82.666667	pass
<b>18</b>	19	5	89	68	87	244	81.333333	pass
<b>19</b>	20	5	78	83	58	219	73.000000	pass

In [ ]:

## [실습 6-2] 'nclass'별 인원 수 구하기

`new_exam` 데이터 프레임을 사용하여

> 결과 데이터 프레임에 'nclass'와 'count'가 나타나도록 한다.

```
In [18]: ## 'nclass'별 인원 수 구하기
```

```
Out[18]:
```

	nclass	count
0	1	4
1	2	4
2	3	4
3	4	4
4	5	4

```
In [ ]:
```

### [실습 6-3] 원하는 통계 결과만 필터링 하여 구하기

new\_exam 데이터 프레임을 사용하여

> 'nclass'별 'mean'의 평균을 구하여, 평균 값이 70 이상인 행만 구하기

```
In [19]: ##### > 'nclass'별 'mean'의 평균을 구하여, 평균 값이 70 이상인 행만 구하기
```

```
Out[19]:
```

	nclass	avg
4	5	78.5

```
In [ ]:
```

### [실습 6-4] 데이터 행, 열 분리 및 병합

new\_exam 데이터 프레임을 사용하여

<1> exam\_mean 데이터 프레임 만들기

<1-1> new\_exam 데이터 프레임에서 'id', 'nclass', 'total', 'mean' 변수를 추출하여 exam\_mean 만들기

<1-2> new\_exam 데이터 프레임에서 'id', 'math', 'english', 'science' 변수를 추출하여 exam\_subj 만들기

<2> exam\_mean과 exam\_subj를 id를 기준으로 열 합병하여 exam\_result 데이터 프레임 생성

<3> nclass로 행 분리된 exam\_1class, exam\_2class, exam\_3class, exam\_4class, exam\_5class 프레임 생성

<4> exam\_1class, exam\_2class, exam\_3class, exam\_4class, exam\_5class 프레임들의 모든 행들을 병합하여 exam\_nclass 생성

```
In [22]: ## <1-1> new_exam 데이터 프레임에서 'id', 'nclass', 'total', 'mean' 변수를 추출하여 exam_mean 데이터 프레임 만들기
```

```
exam_mean.head()
```

```
Out[22]:
```

	id	nclass	total	mean
0	1	1	198	66.000000
1	2	1	217	72.333333
2	3	1	209	69.666667
3	4	1	186	62.000000
4	5	2	170	56.666667

```
In [23]: ## <1-2> exam_new 데이터 프레임에서 'id', 'math', 'english', 'science' 변수를 추출하여 exam_subj 데이터 프레임 만들기
```

```
exam_subj.head()
```

```
Out[23]:
```

	id	math	english	science
0	1	50	98	50
1	2	60	97	60
2	3	45	86	78
3	4	30	98	58
4	5	25	80	65

```
In [25]: ## <2> exam_mean과 exam_subj를 id를 기준으로 열 합병하여 exam_result 데이터 프레임 생성
```

```
exam_result.head()
```

Out[25]:

	id	nclass	total	mean	math	english	science
0	1	1	198	66.000000	50	98	50
1	2	1	217	72.333333	60	97	60
2	3	1	209	69.666667	45	86	78
3	4	1	186	62.000000	30	98	58
4	5	2	170	56.666667	25	80	65

In [26]:

```
## <3> exam_result를 nclass를 기준으로 행 분리하여 별도의 데이터 프레임 exam_1class, exam_2class, exam_3class, exam_4class, exam_5c

print(exam_1class)
print(exam_2class)
print(exam_3class)
print(exam_4class)
print(exam_5class)
```

	id	nclass	total	mean	math	english	science
0	1	1	198	66.000000	50	98	50
1	2	1	217	72.333333	60	97	60
2	3	1	209	69.666667	45	86	78
3	4	1	186	62.000000	30	98	58
	id	nclass	total	mean	math	english	science
4	5	2	170	56.666667	25	80	65
5	6	2	237	79.000000	50	89	98
6	7	2	215	71.666667	80	90	45
7	8	2	193	64.333333	90	78	25
	id	nclass	total	mean	math	english	science
8	9	3	133	44.333333	20	98	15
9	10	3	193	64.333333	50	98	45
10	11	3	195	65.000000	65	65	65
11	12	3	162	54.000000	45	85	32
	id	nclass	total	mean	math	english	science
12	13	4	209	69.666667	46	98	65
13	14	4	147	49.000000	48	87	12
14	15	4	209	69.666667	75	56	78
15	16	4	221	73.666667	58	98	65
	id	nclass	total	mean	math	english	science
16	17	5	231	77.000000	65	68	98
17	18	5	248	82.666667	80	78	90
18	19	5	244	81.333333	89	68	87
19	20	5	219	73.000000	78	83	58

In [27]: ## <4> exam\_1class, exam\_2class, exam\_3class, exam\_4class, exam\_5class 프레임들의 모든 행들을 병합하여 exam\_nclass 생성

```
exam_nclass
```

Out[27]:

	id	nclass	total	mean	math	english	science
<b>0</b>	1	1	198	66.000000	50	98	50
<b>1</b>	2	1	217	72.333333	60	97	60
<b>2</b>	3	1	209	69.666667	45	86	78
<b>3</b>	4	1	186	62.000000	30	98	58
<b>4</b>	5	2	170	56.666667	25	80	65
<b>5</b>	6	2	237	79.000000	50	89	98
<b>6</b>	7	2	215	71.666667	80	90	45
<b>7</b>	8	2	193	64.333333	90	78	25
<b>8</b>	9	3	133	44.333333	20	98	15
<b>9</b>	10	3	193	64.333333	50	98	45
<b>10</b>	11	3	195	65.000000	65	65	65
<b>11</b>	12	3	162	54.000000	45	85	32
<b>12</b>	13	4	209	69.666667	46	98	65
<b>13</b>	14	4	147	49.000000	48	87	12
<b>14</b>	15	4	209	69.666667	75	56	78
<b>15</b>	16	4	221	73.666667	58	98	65
<b>16</b>	17	5	231	77.000000	65	68	98
<b>17</b>	18	5	248	82.666667	80	78	90
<b>18</b>	19	5	244	81.333333	89	68	87
<b>19</b>	20	5	219	73.000000	78	83	58

In [ ]: