

## [ 판다스 실습(3) ]

(1) "./data/score.csv"파일을 읽고 다음 결과를 출력해 본다.

```
컬럼명 : Index(['name', 'kor', 'eng', 'mat'], dtype='object')
인덱스 : RangeIndex(start=0, stop=5, step=1)
```

	name	kor	eng	mat	total	avg
0	둘리	90	90	90	270	90.00
1	또치	80	80	80	240	80.00
2	도우너	70	70	70	210	70.00
3	희동이	90	80	70	240	80.00
4	마이클	60	50	50	160	53.33

(2) "./data/score\_noheader.csv"파일을 읽고 다음 결과를 출력해 본다.

	0	1	2	3
0	둘리	90	90	90
1	또치	80	80	80
2	도우너	70	70	70
3	희동이	90	80	70
4	마이클	60	50	50

```
컬럼명 : Int64Index([0, 1, 2, 3], dtype='int64')
인덱스 : RangeIndex(start=0, stop=5, step=1)
컬럼명 추가 -----
컬럼명 : Index(['name', 'kor', 'eng', 'mat'], dtype='object')
```

	name	kor	eng	mat
0	둘리	90	90	90
1	또치	80	80	80
2	도우너	70	70	70
3	희동이	90	80	70
4	마이클	60	50	50

total과 avg열 추가 -----

	name	kor	eng	mat	total	avg
0	둘리	90	90	90	270	90.00
1	또치	80	80	80	240	80.00
2	도우너	70	70	70	210	70.00
3	희동이	90	80	70	240	80.00
4	마이클	60	50	50	160	53.33

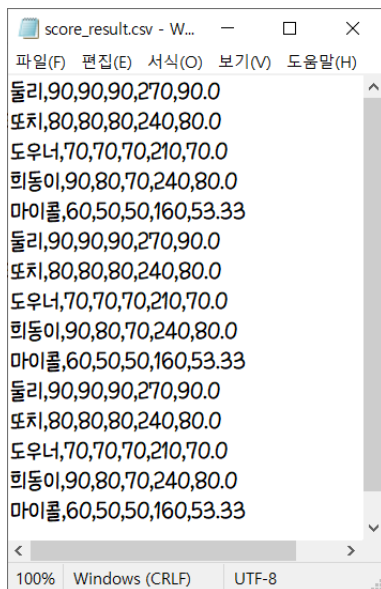
(3) "./data/score\_header.csv" 파일을 읽고 다음 결과를 출력해 본다.  
또한 "output/score\_result.csv" 파일에 **추가모드**로 다음에 제시된 결

과와 같이 출력하는데 헤더와 인덱스는 출력하지 않는다.

	name	kor	eng	mat
0	둘리	90	90	90
1	또치	80	80	80
2	도우너	70	70	70
3	희동이	90	80	70
4	마이클	60	50	50

컬럼명 : Index(['name', 'kor', 'eng', 'mat'], dtype='object')  
인덱스 : RangeIndex(start=0, stop=5, step=1)

	name	kor	eng	mat	total	avg
0	둘리	90	90	90	270	90.00
1	또치	80	80	80	240	80.00
2	도우너	70	70	70	210	70.00
3	희동이	90	80	70	240	80.00
4	마이클	60	50	50	160	53.33



(4) "./data/score.xlsx" 파일을 읽고 다음 결과를 출력해 본다.

또한 "output/score\_result1.xlsx" 과 "output/score\_result2.xlsx" 파일에 출력하는데 두 개 파일 모두 헤더가 출력되게 하며 score\_result1.xlsx 는 인덱스를 출력하고 score\_result2.xlsx 는 인덱스를 출력하지 않는다.

	name	kor	eng	mat
0	둘리	90	90	90
1	또치	80	80	80
2	도우너	70	70	70
3	희동이	90	80	70
4	마이클	60	50	50
5	고길동	90	90	0

	name	kor	eng	mat	total	avg
0	둘리	90	90	90	270	90.00
1	또치	80	80	80	240	80.00
3	희동이	90	80	70	240	80.00
2	도우너	70	70	70	210	70.00
5	고길동	90	90	0	180	60.00
4	마이클	60	50	50	160	53.33

	name	kor	eng	mat	total	avg
1	둘리	90	90	90	270	90
2	또치	80	80	80	240	80
3	희동이	90	80	70	240	80
4	도우너	70	70	70	210	70
5	고길동	90	90	0	180	60
6	마이클	60	50	50	160	53.33

	name	kor	eng	mat	total	avg
0	둘리	90	90	90	270	90
1	또치	80	80	80	240	80
2	희동이	90	80	70	240	80
3	도우너	70	70	70	210	70
4	고길동	90	90	0	180	60
5	마이클	60	50	50	160	53.33

(5) "data/mydata.json" 파일을 읽고 다음 결과를 출력해 본다.

기본정보 보기

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 4 columns):
#   Column  Non-Null Count  Dtype
---  -
0   year    24 non-null      int64
1   item1   24 non-null      int64
2   item2   24 non-null      int64
3   item3   24 non-null      int64
dtypes: int64(4)
memory usage: 896.0 bytes
None
앞에서 부터 5개만 미리 보기
```

	year	item1	item2	item3
0	1990	95	20	15
1	1991	65	10	35
2	1992	45	30	90
3	1993	10	40	70
4	1994	22	50	50

뒤에서 부터 5개만 미리 보기

	year	item1	item2	item3
19	2009	99	90	70
20	2010	75	85	45
21	2011	22	42	22
22	2012	63	13	30
23	2013	80	40	90

앞에서 부터 10개만 미리 보기

	year	item1	item2	item3
0	1990	95	20	15
1	1991	65	10	35
2	1992	45	30	90
3	1993	10	40	70
4	1994	22	50	50
5	1995	35	70	30
6	1996	40	80	25
7	1997	25	90	75
8	1998	15	57	95
9	1999	45	79	33

(6)

(24, 4)

행의 개수 : 24, 열의 개수 : 4

"/data/mpgdata.csv" 파일을 읽고 다음 결과를 출력해 본다.

앞에서 부터 3개만 미리 보기

	mpg	cylinders	displacement	horsepower	weight	acceleration	model-year
0	18.0	8	307.0	130.0	3504	12.0	70
1	15.0	8	350.0	165.0	3693	11.5	70
2	18.0	8	318.0	150.0	3436	11.0	70

(398, 7)  
행의 개수 398  
열의 개수 7  
기술통계 정보 요약

	mpg	cylinders	displacement	horsepower	weight	acceleration	model-year
count	398.000000	398.000000	398.000000	396.000000	398.000000	398.000000	398.000000
mean	23.514573	5.454774	193.425879	104.189394	2970.424623	15.568090	76.010050
std	7.815984	1.701004	104.269838	38.402030	846.841774	2.757689	3.697627
min	9.000000	3.000000	68.000000	46.000000	1613.000000	8.000000	70.000000
25%	17.500000	4.000000	104.250000	75.000000	2223.750000	13.825000	73.000000
50%	23.000000	4.000000	148.500000	92.000000	2803.500000	15.500000	76.000000
75%	29.000000	8.000000	262.000000	125.000000	3608.000000	17.175000	79.000000
max	46.600000	8.000000	455.000000	230.000000	5140.000000	24.800000	82.000000