

데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

데이터 전처리 연습문제 풀이

문제1

패키지 참조

```
from pandas import DataFrame
```

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

딕셔너리의 key가 column 이름이 된다.

```
data = {
    "class": ["A1", "A1", "A1", "A1", "A1",
              "A2", "A2", "A2", "A2", "A2", "A2",
              "A3", "A3", "A3", "A3", "A3", "A3", "A3",
              "A4", "A4", "A4", "A4", "A4", "A4", "A4"],
    "time": [62, 60, 63, 59, 61,
              62, 67, 71, 64, 65, 66,
              68, 66, 71, 67, 68, 69, 70,
              58, 62, 60, 59, 61, 63, 64]
}
```

데이터 전처리 연습문제 폴이

문제1

패키지 참조

폴이1) 리스트를 원소로 갖는 딕셔너리 활용

폴이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

```
df = DataFrame(data)
df
```

	class	time
0	A1	62
1	A1	60
2	A1	63
3	A1	59
4	A1	61
5	A2	62
6	A2	67
7	A2	71
8	A2	64
9	A2	65
10	A2	66
11	A3	68
12	A3	66
13	A3	71
14	A3	67
15	A3	68

데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

	class	time
16	A3	69
17	A3	70
18	A4	58
19	A4	62
20	A4	60
21	A4	59
22	A4	61
23	A4	63
24	A4	64

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

```
data = [
    {"class": "A1", "time": 62},
    {"class": "A1", "time": 60},
    {"class": "A1", "time": 63},
    {"class": "A1", "time": 59},
    {"class": "A1", "time": 61},
    {"class": "A2", "time": 62},
    {"class": "A2", "time": 67},
    {"class": "A2", "time": 71},
    {"class": "A2", "time": 64},
    {"class": "A2", "time": 65},
```

데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

```
{
  "class": "A2", "time": 66},
  {"class": "A3", "time": 68},
  {"class": "A3", "time": 66},
  {"class": "A3", "time": 71},
  {"class": "A3", "time": 67},
  {"class": "A3", "time": 68},
  {"class": "A3", "time": 69},
  {"class": "A3", "time": 70},
  {"class": "A4", "time": 58},
  {"class": "A4", "time": 62},
  {"class": "A4", "time": 60},
  {"class": "A4", "time": 59},
  {"class": "A4", "time": 61},
  {"class": "A4", "time": 63},
  {"class": "A4", "time": 64}
}
```

]

df = DataFrame(data)

df

	class	time
0	A1	62
1	A1	60
2	A1	63
3	A1	59
4	A1	61

데이터 전처리 연습문제 폴이

문제1

패키지 참조

폴이1) 리스트를 원소로 갖는 딕셔너리 활용

폴이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

	class	time
5	A2	62
6	A2	67
7	A2	71
8	A2	64
9	A2	65
10	A2	66
11	A3	68
12	A3	66
13	A3	71
14	A3	67
15	A3	68
16	A3	69
17	A3	70
18	A4	58
19	A4	62
20	A4	60
21	A4	59
22	A4	61

데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

	class	time
23	A4	63
24	A4	64

class를 범주형으로 변환

생성된 데이터 프레임에 대한 각 변수의 데이터 타입 확인

```
df.dtypes
```

```
class      object
time       int64
dtype: object
```

데이터프레임의 여러 컬럼을 일괄 변경하여 새로운 데이터프레임으로 생성

```
df2 = df.astype({'class': 'category'})
df2.dtypes
```

```
class      category
time       int64
dtype: object
```

특정 컬럼의 타입만 변경 후 원본에 바로 적용

데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

```
df['class'] = df['class'].astype('category')
df.dtypes
```

```
class      category
time      int64
dtype: object
```

서열척도로 변경해야 하는 경우

```
from pandas import CategoricalDtype
df['class'] = df['class'].astype(
    CategoricalDtype(categories=['A1', 'A2', 'A3', 'A4'], ordered=True)
df.dtypes
```

```
class      category
time      int64
dtype: object
```

문제2

```
df1 = DataFrame([
    {"ID": 25, "READ": 350, "MATH": 425, "GRADE": 7},
    {"ID": 27, "READ": 425, "MATH": 375, "GRADE": 8},
    {"ID": 30, "READ": 475, "MATH": 485, "GRADE": 8},
```

데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

```

{"ID": 35, "READ": 375, "MATH": 400, "GRADE": 9}
])
df1

```

	ID	READ	MATH	GRADE
0	25	350	425	7
1	27	425	375	8
2	30	475	485	8
3	35	375	400	9

```

df2 = DataFrame([
    {"ID": 20, "READ": 250, "MATH": 300, "GRADE": 7},
    {"ID": 33, "READ": 325, "MATH": 225, "GRADE": 9},
    {"ID": 37, "READ": 300, "MATH": 375, "GRADE": 9}
])
df2

```

	ID	READ	MATH	GRADE
0	20	250	300	7
1	33	325	225	9
2	37	300	375	9

```

from pandas import concat

```


데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

```
df = concat([df1, df2], ignore_index=True)
df
```

	ID	READ	MATH	GRADE
0	25	350	425	7
1	27	425	375	8
2	30	475	485	8
3	35	375	400	9
4	20	250	300	7
5	33	325	225	9
6	37	300	375	9

문제3

```
from pandas import read_excel, pivot_table
```

```
df = read_excel("https://data.hossam.kr/C02/smoke.xlsx")
df
```

	지역	구성	흡연율
0	도시	남	0.64

데이터 전처리 연습문제 폴이

문제1

패키지 참조

폴이1) 리스트를 원소로 갖는 딕셔너리 활용

폴이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

	지역	구성	흡연율
1	도시	여	0.45
2	농촌	공학	0.70
3	도시	남	0.85
4	농촌	남	0.72
5	도시	남	0.78
6	도시	여	0.62
7	도시	남	0.79
8	농촌	남	0.75
9	농촌	공학	0.81
10	농촌	여	0.56
11	도시	여	0.48
12	농촌	남	0.72
13	도시	공학	0.68
14	농촌	남	0.79
15	도시	남	0.90
16	농촌	여	0.34
17	도시	여	0.48
18	농촌	여	0.49

데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

	지역	구성	흡연율
19	농촌	공학	0.76

```

pivot_table(df,
             index="지역",
             columns="구성",
             values="흡연율",
             aggfunc="mean")

```

구성	공학	남	여
지역			
농촌	0.756667	0.745	0.463333
도시	0.680000	0.792	0.507500

```

pivot_table(df,
             index="구성",
             columns="지역",
             values="흡연율",
             aggfunc="mean")

```

지역	농촌	도시
구성		
공학	0.756667	0.6800

데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

지역	농촌	도시
구성		
남	0.745000	0.7920
여	0.463333	0.5075

문제4

```
df = read_excel("https://data.hossam.kr/C02/privacy.xlsx")
df
```

	학생ID	성별	메모종류
0	1	남	3
1	2	여	2
2	3	여	2
3	4	여	1
4	5	여	2
5	6	여	3
6	7	남	2
7	8	여	2
8	9	여	1

데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

	학생ID	성별	메모종류
9	10	남	1
10	11	남	2
11	12	남	3
12	13	남	2
13	14	남	1
14	15	남	2
15	16	여	2
16	17	여	1
17	18	여	2
18	19	여	3
19	20	여	2
20	21	여	2
21	22	여	2
22	23	여	2
23	24	여	2
24	25	여	2
25	26	여	2
26	27	여	2

데이터 전처리 연습문제 풀이

문제1

패키지 참조

풀이1) 리스트를 원소로 갖는 딕셔너리 활용

풀이2) 딕셔너리를 원소로 갖는 리스트 활용

class를 범주형으로 변환

문제2

문제3

문제4

	학생ID	성별	메모종류
27	28	여	2
28	29	여	1
29	30	여	2

```
from pandas import crosstab
```

```
crosstab(index=df['성별'], columns=df['메모종류'], normalize=True, margins=True)
```

메모종류	1	2	3	All
성별				
남	0.066667	0.133333	0.066667	0.266667
여	0.133333	0.533333	0.066667	0.733333
All	0.200000	0.666667	0.133333	1.000000