

## <5장> 통계 분석

# 학습 목표

- 기술 통계(요약 통계)를 이용해 기본적인 통계 분석 방법을 익힌다.
- 독립변수와 종속변수 간의 상호 연관성 정도를 파악하기 위한 회귀분석 방법을 익힌다.
- 데이터에서 찾은 평균 값으로 두 그룹 사이에 차이가 있는지 확인하는 t-검정을 익힌다.
- 데이터 사이에 어떤 선형적 관계가 있는지를 분석하는 상관관계 분석을 익힌다.
- 기술 통계 및 상관관계 분석 결과를 시각화하여 분석 결과를 해석한다.

# 목차

01 기술 통계 분석과 시각화

02 상관관계 분석과 시각화

03 통계분석 실습(happy\_merge.csv)

01

# 기술 통계 분석과 시각화

# 1. 기술통계분석과 시각화

## ■ 기술통계 분석(Descriptive Statistics Analysis)

- 데이터의 특성을 요약하고 설명하는 통계 기법
- 복잡한 데이터를 단순한 수치나 그래프로 정리해서 전체적인 경향이나 분포를 한눈에 파악할 수 있도록 도와 줌

## ■ 주요 구성요소

- 중심 경향치:
  - 평균(Mean): 데이터의 중심값
  - 중앙값(Median): 정렬된 데이터의 가운데 값
  - 최빈값(Mode): 가장 자주 나타나는 값
- 산포도(흩어짐 정도):
  - 범위(Range): 최대값 - 최소값
  - 분산(Variance), 표준편차(Standard Deviation): 데이터가 평균에서 얼마나 퍼져 있는지
- 분포의 형태:
  - 왜도(Skewness): 데이터의 비대칭성
  - 첨도(Kurtosis): 데이터의 뾰족함 정도
- 표나 그래프:
  - 빈도표, 히스토그램, 막대그래프, 상자그림(box plot) 등을 활용하여 시각적으로 요약

# 1. 기술통계분석과 시각화

## ■ 회귀 분석(Regression Analysis)

- 변수 간의 관계를 분석하고 예측하기 위한 통계 기법
- 한 변수(종속변수, Y)가 다른 변수들(독립변수, X)에 따라 어떻게 변하는지를 분석
- 회귀 분석 식:  $y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$
- 회귀분석 활용
  - 예측(Prediction): 집값, 매출, 시험점수 등 미래 값을 예측
  - 설명(Explanation): 어떤 변수가 결과에 영향을 주는지 파악
  - 인과추론(Causality): 변수 간 인과관계를 모형화
  - 로지스틱 회귀(Logistic Regression)
    - 종속변수가 범주형일 때 사용
    - (예: 합격/불합격)출력은 확률, 결과는 0 또는 1 (분류 문제)
  - 로지스틱 회귀(Logistic Regression)
    - 종속변수가 범주형일 때 사용

# . 기술통계분석과 시각화

## ■ 회귀분석

### ■ 대표적인 회귀 분석 종류

- 단순 선형 회귀(Simple Linear Regression)
  - 하나의 독립변수  $X$ 와 종속변수  $Y$ 의 직선 관계 모델링
  - 예: 키( $X$ )  $\rightarrow$  몸무게( $Y$ )
  - 수식:  $Y = \beta_0 + \beta_1 X + \varepsilon$
- 다중 회귀(Multiple Linear Regression)
  - 여러 개의 독립변수  $X_1, X_2, \dots$ 으로  $Y$  예측
  - 예: 공부시간, 수면시간  $\rightarrow$  시험점수

### ■ 회귀분석 결과에서 제공되는 정보

- 계수( $\beta$ ): 변수의 영향력
- $R^2$  (결정계수): 모델 설명력 (0~1 사이 값)
- p-value: 변수의 통계적 유의성

# 1. 기술통계분석과 시각화

## ■ t-검정(t-test)

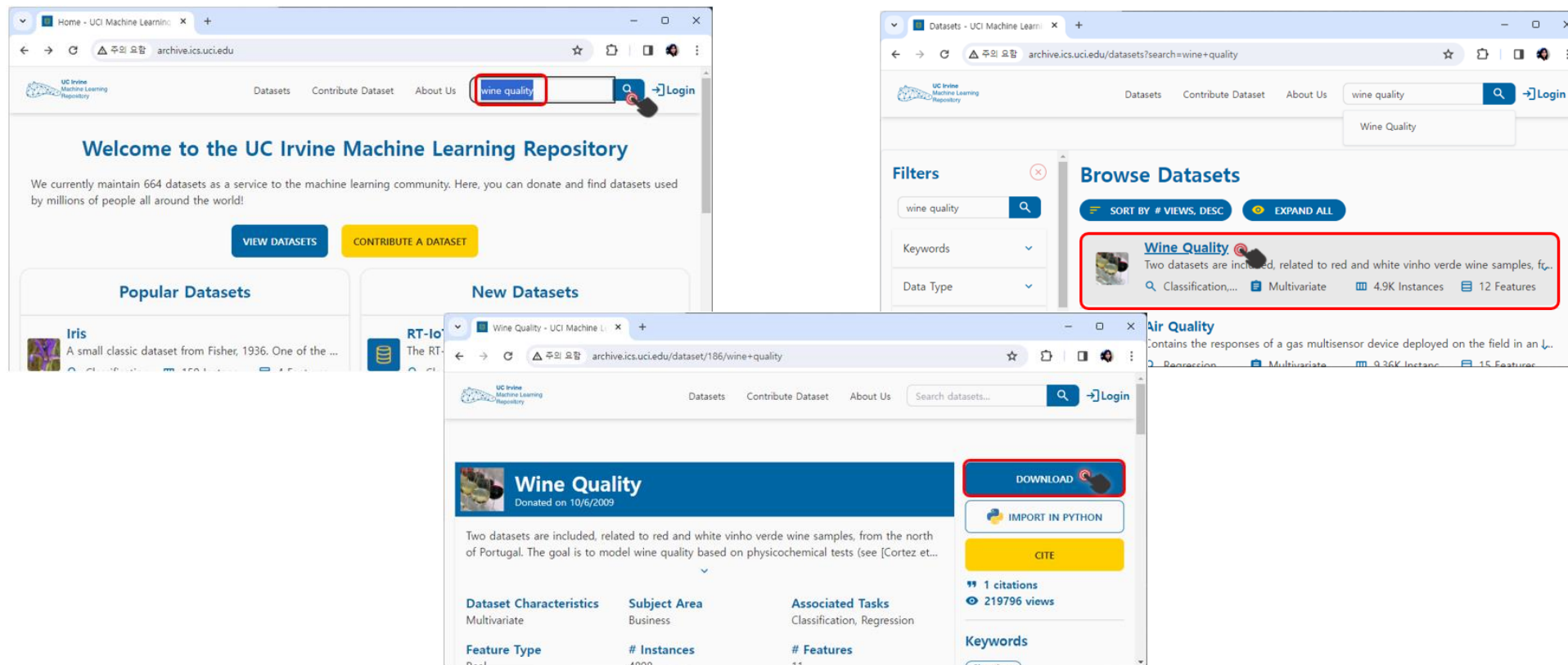
- 두 그룹의 평균을 비교하여 차이가 통계적으로 유의미한지를 판단하는 통계 기법
- 예) 남녀의 키 차이, 실험군과 대조군의 치료 효과 차이 등
- 대표적인 t-검정 종류:
  - 독립표본 t-검정 (Independent t-test)
    - 서로 다른 두 집단의 평균 비교
    - 예: A반과 B반의 시험 성적 차이
  - 대응표본 t-검정 (Paired t-test)
    - 같은 집단의 전후 비교
    - 예: 운동 전·후 체중 변화
  - 단일표본 t-검정 (One-sample t-test)
    - 하나의 그룹 평균이 특정 값과 다른지 비교
    - 예: 특정 반의 평균 점수가 70점과 유의미하게 다른가?
- 해석 방법:
  - t값: 두 평균 간의 차이 크기
  - p값:  $p < 0.05$ : 유의미한 차이 있음
  - $p \geq 0.05$ : 유의미한 차이 없음



# 1. 기술통계분석과 시각화

## ■ 데이터 수집

- 캘리포니아 어바인 대학의 머신러닝 저장소에서 제공하는 오픈 데이터 사용( <https://archive.ics.uci.edu/> )
  - wine quality 검색
  - 다운로드한 wine+quality.zip 파일을 압축을 해제한 후 winequality-red.csv, winequality-white.csv 파일을 data 폴더에 저장

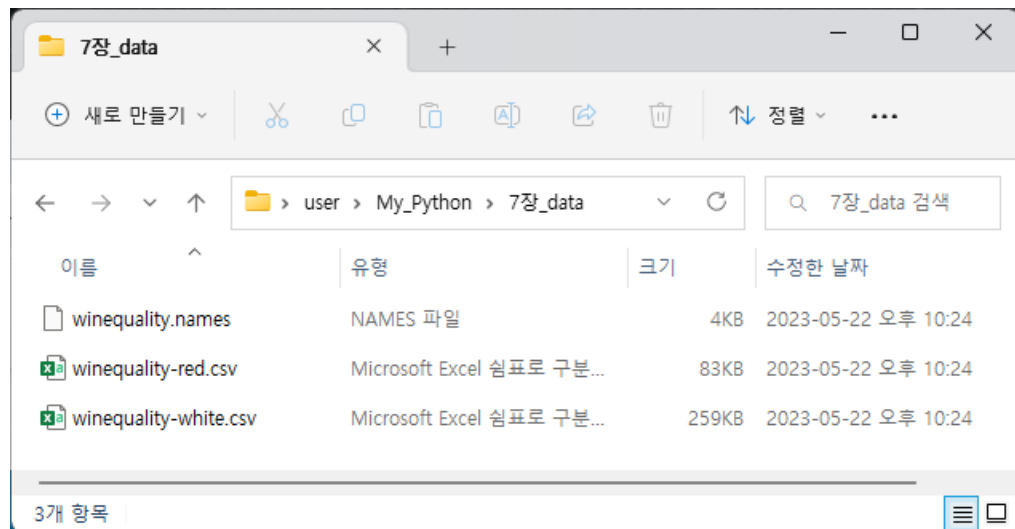


# 1. 데이터 시각화 이해

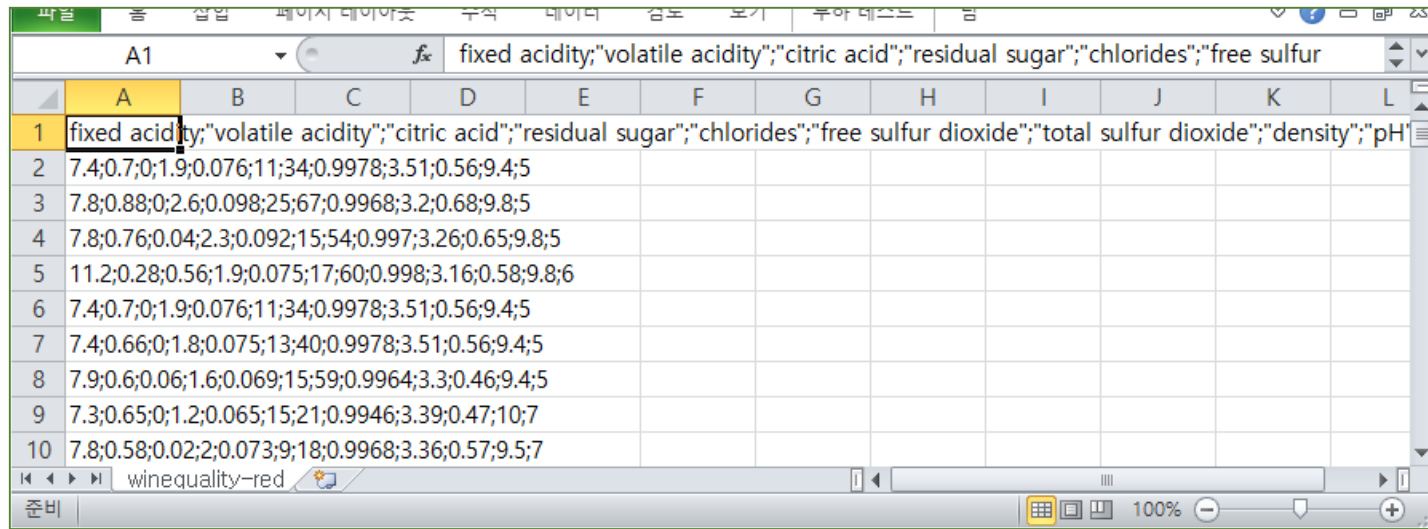
## ■ 데이터 준비

### 1. 다운로드한 CSV 파일 정리하기

- 엑셀은 CSV 파일을 열 때 쉼표를 열 구분자로 사용하므로 열이 깨진 것처럼 보임



(a) 압축을 풀고 저장한 파일 목록



(b) csv 데이터 파일

# 1. 데이터 시각화 이해

## ■ 데이터 준비

### 1. 다운로드한 CSV 파일 정리하기

#### 1) 엑셀에서 열 구분자를 세미콜론으로 인식시키기

#1. 엑셀에서 열 구분자를 세미콜론으로 인식시키기 -----

```
import pandas as pd
```

```
red_df = pd.read_csv('data/winequality-red.csv', sep = ';', header = 0)
```

```
white_df = pd.read_csv('data/winequality-white.csv', sep = ';', header = 0)
```

```
red_df.to_csv('data/winequality-red2.csv', index = False)
```

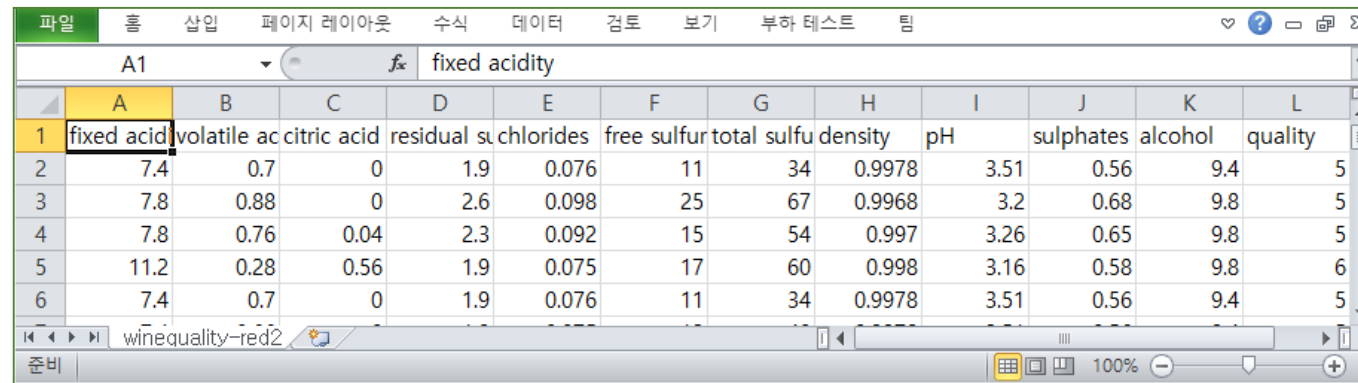
```
white_df.to_csv('data/winequality-white2.csv', index = False)
```

# 1. 데이터 시각화 이해

## ■ 데이터 준비

### 1. 다운로드한 CSV 파일 정리하기

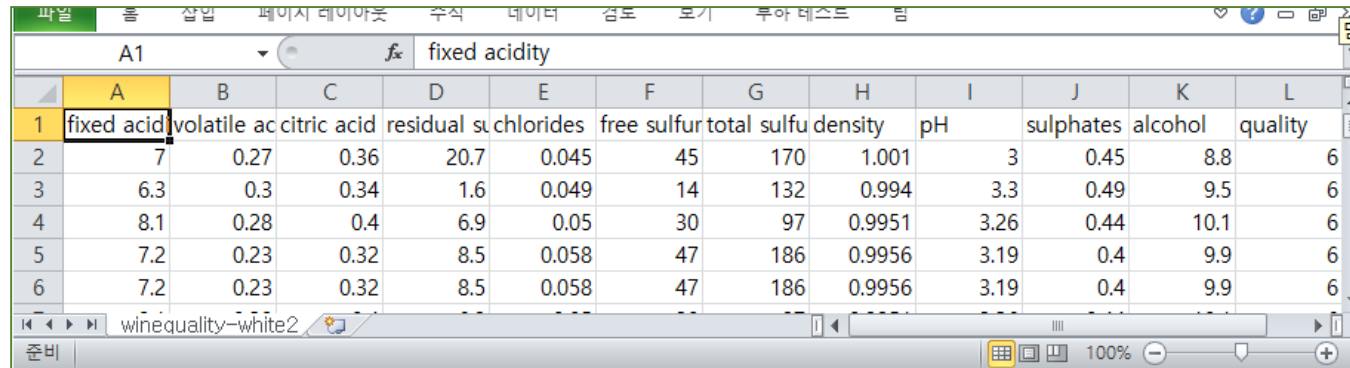
2) 파이썬에서 다시 저장한 winequality-red2.csv와 winequality-white2.csv 파일을 엑셀에서 다시 열어서 확인



This screenshot shows an Excel spreadsheet for the file 'winequality-red2.csv'. The spreadsheet has 13 columns labeled A through L. Column A contains the header 'fixed acidity'. Column B contains 'volatile acidity', C contains 'citric acid', D contains 'residual sugar', E contains 'chlorides', F contains 'free sulfur dioxide', G contains 'total sulfur dioxide', H contains 'density', I contains 'pH', J contains 'sulphates', K contains 'alcohol', and L contains 'quality'. The data rows show values for these attributes, with the 'quality' column ranging from 5 to 6.

	A	B	C	D	E	F	G	H	I	J	K	L
1	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
2	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
3	7.8	0.88	0	2.6	0.098	25	67	0.9968	3.2	0.68	9.8	5
4	7.8	0.76	0.04	2.3	0.092	15	54	0.997	3.26	0.65	9.8	5
5	11.2	0.28	0.56	1.9	0.075	17	60	0.998	3.16	0.58	9.8	6
6	7.4	0.7	0	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5

(a) winequality-red2.csv



This screenshot shows an Excel spreadsheet for the file 'winequality-white2.csv'. The spreadsheet has 13 columns labeled A through L. Column A contains the header 'fixed acidity'. Column B contains 'volatile acidity', C contains 'citric acid', D contains 'residual sugar', E contains 'chlorides', F contains 'free sulfur dioxide', G contains 'total sulfur dioxide', H contains 'density', I contains 'pH', J contains 'sulphates', K contains 'alcohol', and L contains 'quality'. The data rows show values for these attributes, with the 'quality' column ranging from 6 to 10.

	A	B	C	D	E	F	G	H	I	J	K	L
1	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
2	7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
3	6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
4	8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
5	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
6	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6

(b) winequality-white2.csv

# 1. 데이터 시각화 이해

## ■ 데이터 준비

### 1. 다운로드한 CSV 파일 정리하기

- 데이터 설명
  - 머신러닝 입력용으로 제공하는 데이터셋으로 작성한 것이므로 데이터 정제 등의 전처리 작업은 이미 되어 있음
  - winequality-red2.csv : 레드 와인에 대한 데이터셋.
    - » 행이 1,599개이므로 샘플이 1,599개.
    - » 열은 12개. 머신러닝용 데이터셋은  $n$ 개 속성 중에서 마지막 속성이 출력 변수( $y$ )이고, 앞서서부터  $n-1$ 개 속성은 입력 변수( $x$ ).
    - » 입력 변수 : fixed acidity(고정산), volatile acidity(휘발산), citric acid(구연산), residual sugar(잔당), chlorides(염화물), free sulfur dioxide(유리 이산화황), total sulfur dioxide(총 이산화황), density(밀도), pH, sulphates(황산염), alcohol(알코올) 등의 11개 속성
    - » 출력 변수 : 와인의 품질 등급을 나타내는 quality.
  - winequality-white2.csv : 화이트 와인에 대한 데이터셋.
    - » 행이 4,898개이므로 샘플이 4,898개
    - » 열은 winequality-red2.csv와 마찬가지로 11개의 입력 변수와 1개의 출력 변수 quality로 구성되어 있음

# 1. 데이터 시각화 이해

## ■ 데이터 준비

### 2. 데이터 병합하기

#### 1) 레드 와인과 화이트 와인 파일 합치기 - 와인 종류 구분을 위해 type 컬럼 추가\_insert()

(1) 레드 와인 데이터 확인 및 정리

```
01 >>> red_df.head()
   fixed acidity volatile acidity  citric acid ... sulphates  alcohol  quality
0         7.4             0.70         0.00 ...         0.56         9.4         5
1         7.8             0.88         0.00 ...         0.68         9.8         5
2         7.8             0.76         0.04 ...         0.65         9.8         5
3        11.2             0.28         0.56 ...         0.58         9.8         6
4         7.4             0.70         0.00 ...         0.56         9.4         5

[5 rows x 12 columns]
02 >>> red_df.insert(0, column = 'type', value = 'red')
03 >>> red_df.head()
   type  fixed acidity  volatile acidity ... sulphates  alcohol  quality
0  red         7.4             0.70 ...         0.56         9.4         5
1  red         7.8             0.88 ...         0.68         9.8         5
2  red         7.8             0.76 ...         0.65         9.8         5
3  red        11.2             0.28 ...         0.58         9.8         6
4  red         7.4             0.70 ...         0.56         9.4         5

[5 rows x 13 columns]
04 >>> red_df.shape
(1599, 13)
```

# 1. 데이터 시각화 이해

## ■ 데이터 준비

### 2. 데이터 병합하기

1) 레드 와인과 화이트 와인 파일 합치기 - 와인 종류 구분을 위해 type 컬럼 추가\_insert()

(2) 화이트 와인 데이터 확인 및 정리

```
05 >>> white_df.head()
   fixed acidity  volatile acidity  citric acid  ...  sulphates  alcohol  quality
0         7.0             0.27         0.36  ...      0.45        8.8         6
1         6.3             0.30         0.34  ...      0.49        9.5         6
2         8.1             0.28         0.40  ...      0.44       10.1         6
3         7.2             0.23         0.32  ...      0.40        9.9         6
4         7.2             0.23         0.32  ...      0.40        9.9         6

[5 rows x 12 columns]
06 >>> white_df.insert(0, column = 'type', value = 'white')
07 >>> white_df.head()
   type  fixed acidity  volatile acidity  ...  sulphates  alcohol  quality
0  white         7.0             0.27  ...      0.45        8.8         6
1  white         6.3             0.30  ...      0.49        9.5         6
2  white         8.1             0.28  ...      0.44       10.1         6
3  white         7.2             0.23  ...      0.40        9.9         6
4  white         7.2             0.23  ...      0.40        9.9         6

[5 rows x 13 columns]
08 >>> white_df.shape
(4898, 13)
```

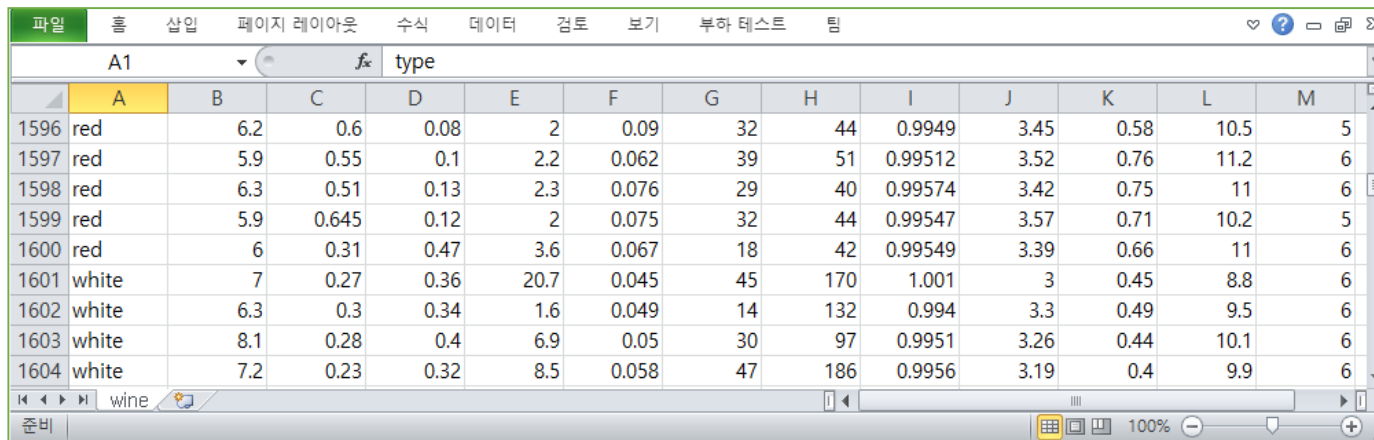
# 1. 데이터 시각화 이해

## ■ 데이터 준비

### 2. 데이터 병합하기

- 1) 레드 와인과 화이트 와인 파일 합치기 - 와인 종류 구분을 위해 type 컬럼 추가\_insert()
- (3) 레드 와인과 화이트 와인 파일 합치기\_concat()

```
09 >>> wine = pd.concat([red_df, white_df])
10 >>> wine.shape
    (6497, 13)
11 >>> wine.to_csv('data/wine.csv', index = False)
```



	A	B	C	D	E	F	G	H	I	J	K	L	M
1596	red	6.2	0.6	0.08	2	0.09	32	44	0.9949	3.45	0.58	10.5	5
1597	red	5.9	0.55	0.1	2.2	0.062	39	51	0.99512	3.52	0.76	11.2	6
1598	red	6.3	0.51	0.13	2.3	0.076	29	40	0.99574	3.42	0.75	11	6
1599	red	5.9	0.645	0.12	2	0.075	32	44	0.99547	3.57	0.71	10.2	5
1600	red	6	0.31	0.47	3.6	0.067	18	42	0.99549	3.39	0.66	11	6
1601	white	7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
1602	white	6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
1603	white	8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
1604	white	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6

그림 7-4 레드 와인 데이터셋과 화이트 와인 데이터셋이 결합된 wine.csv 파일



# 1. 데이터 시각화 이해

## ■ 데이터 탐색

### 1. 기본 정보 확인하기 `info()`

```
01 >>> print(wine.info())
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6497 entries, 0 to 4897
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   type                  6497 non-null   object
1   fixed acidity         6497 non-null   float64
2   volatile acidity      6497 non-null   float64
3   citric acid           6497 non-null   float64
4   residual sugar        6497 non-null   float64
5   chlorides             6497 non-null   float64
6   free sulfur dioxide    6497 non-null   float64
7   total sulfur dioxide   6497 non-null   float64
8   density               6497 non-null   float64
9   pH                   6497 non-null   float64
10  sulphates             6497 non-null   float64
11  alcohol               6497 non-null   float64
12  quality               6497 non-null   int64
dtypes: float64(11), int64(1), object(1)
memory usage: 710.6+ KB
None
```

# 1. 데이터 시각화 이해

## ■ 데이터 탐색

### 2. 함수를 사용해 기술 통계 구하기

```
01 >>> wine.columns = wine.columns.str.replace(' ', '_')
```

```
02 >>> wine.head()
```

	type	fixed_acidity	volatile_acidity	...	sulphates	alcohol	quality
0	red	7.4	0.70	...	0.56	9.4	5
1	red	7.8	0.88	...	0.68	9.8	5
2	red	7.8	0.76	...	0.65	9.8	5
3	red	11.2	0.28	...	0.58	9.8	6
4	red	7.4	0.70	...	0.56	9.4	5

[5 rows x 13 columns]

```
03 >>> wine.describe()
```

	fixed_acidity	volatile_acidity	...	alcohol	quality
count	6497.000000	6497.000000	...	6497.000000	6497.000000
mean	7.215307	0.339666	...	10.491801	5.818378
std	1.296434	0.164636	...	1.192712	0.873255
min	3.800000	0.080000	...	8.000000	3.000000
25%	6.400000	0.230000	...	9.500000	5.000000
50%	7.000000	0.290000	...	10.300000	6.000000
75%	7.700000	0.400000	...	11.300000	6.000000
max	15.900000	1.580000	...	14.900000	9.000000

[8 rows x 12 columns]

# 1. 데이터 시각화 이해

## ■ 데이터 탐색

### 2. 함수를 사용해 기술 통계 구하기

```
04 >>> sorted(wine.quality.unique())
[3, 4, 5, 6, 7, 8, 9]
05 >>> wine.quality.value_counts()
6    2836
5    2138
7    1079
4     216
8     193
3      30
9       5
Name: quality, dtype: int64
```

# 1. 데이터 시각화 이해

## ■ 데이터 모델링

### 1. describe() 함수로 그룹 비교하기

```
01 >>> wine.groupby('type')['quality'].describe()
           count    mean      std  min  25%  50%  75%  max
type
Red    1599.0    5.636023  0.807569  3.0   5.0   6.0   6.0   8.0
White  4898.0    5.877909  0.885639  3.0   5.0   6.0   6.0   9.0

02 >>> wine.groupby('type')['quality'].mean()
type
red      5.636023
white    5.877909
Name: quality, dtype: float64

03 >>> wine.groupby('type')['quality'].std()
type
red      0.807569
white    0.885639
Name: quality, dtype: float64

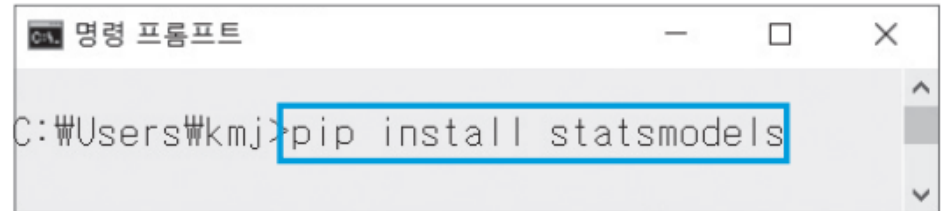
04 >>> wine.groupby('type')['quality'].agg(['mean', 'std'])
           mean      std
type
red    5.636023  0.807569
white  5.877909  0.885639
```

# 1. 데이터 시각화 이해

## ■ 데이터 모델링

### 2. t-검정과 회귀 분석으로 그룹 비교하기

- t-검정을 위해서는 scipy 라이브러리 패키지를 사용
- 회귀 분석을 위해서는 statsmodels 라이브러리 패키지를 사용
- 명령 프롬프트 창에서 다음과 같이 입력하여 statsmodels 패키지 설치



```
C:\Users\Wkmj>pip install statsmodels
```

# 6.2 t-검정과 회귀 분석으로 그룹 비교하기

### cmd창에서 통계패키지 설치하기: pip install statsmodels

```
from scipy import stats
```

```
from statsmodels.formula.api import ols, glm
```

```
red_wine_quality = wine.loc[wine['type'] == 'red', 'quality']
```

```
white_wine_quality = wine.loc[wine['type'] == 'white', 'quality']
```

```
stats.ttest_ind(red_wine_quality, white_wine_quality, equal_var = False)
```

```
Rformula = 'quality ~ fixed_acidity + volatile_acidity + citric_acid + residual_sugar + chlorides + free_sulfur_dioxide  
+ total_sulfur_dioxide + density + pH + sulphates + alcohol'
```

```
regression_result = ols(Rformula, data = wine).fit()
```

```
regression_result.summary()
```

# 1. 데이터 시각화 이해

## ■ 데이터 모델링

### 2. t-검정과 회귀 분석으로 그룹 비교하기

```
08 >>> regression_result.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
```

#### OLS Regression Results

종속 변수	Dep. Variable:	quality	R-squared:	0.292	결정 계수: 모델의 설명력
회귀 모델 종류	Model:	OLS	Adj. R-squared:	0.291	
회귀 분석 함수	Method:	Least Squares	F-statistic:	243.3	모델 전체의 통계적 유의성
	Date:	Tue, 28 Apr 2020	Prob (F-statistic):	0.00	
회귀 분석 수행 시간	Time:	16:19:39	Log-Likelihood:	-7215.5	
샘플 개수: 총 표본 수	No. Observations:	6497	AIC:	1.445e+04	모델 평가 지표
잔차의 자유도: 총 표본 수 - 종속 변수 개수 - 독립 변수 개수	Df Residuals:	6485	BIC:	1.454e+04	
	Df Model:	11			
모델의 자유도: 독립 변수 개수	Covariance Type:	nonrobust			

	회귀 계수	표준 오차	t-통계량	p-value	회귀 계수의 신뢰 구간(95%)	
	coef	std err	t	P> t	[0.025	0.975]
Intercept	55.7627	11.894	4.688	0.000	32.447	79.079
fixed_acidity	0.0677	0.016	4.346	0.000	0.037	0.098
volatile_acidity	-1.3279	0.077	-17.162	0.000	-1.480	-1.176
citric_acid	-0.1097	0.080	-1.377	0.168	-0.266	0.046
residual_sugar	0.0436	0.005	8.449	0.000	0.033	0.054
chlorides	-0.4837	0.333	-1.454	0.146	-1.136	0.168
free_sulfur_dioxide	0.0060	0.001	7.948	0.000	0.004	0.007
total_sulfur_dioxide	-0.0025	0.000	-8.969	0.000	-0.003	-0.002
density	-54.9669	12.137	-4.529	0.000	-78.760	-31.173
pH	0.4393	0.090	4.861	0.000	0.262	0.616
sulphates	0.7683	0.076	10.092	0.000	0.619	0.917
alcohol	0.2670	0.017	15.963	0.000	0.234	0.300

정규 분포성 검정 지표			
Omnibus:	144.075	Durbin-Watson:	1.646
Prob(Omnibus):	0.000	Jarque-Bera (JB):	324.712
Skew:	-0.006	Prob(JB):	3.09e-71
Kurtosis:	4.095	Cond. No.	2.49e+05

회귀 행렬의 조건 수: 다중공선성 지표

#### Warnings:

[1] Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 2.49e+05. This might indicate that there are strong multicollinearity or other numerical problems.

"""

# 1. 데이터 시각화 이해

## ■ 데이터 모델링

### 2. t-검정과 회귀 분석으로 그룹 비교하기(전체 모델 요약)

항목	해석
R-squared = 0.292	설명력은 약 29.2%로, 전체 품질 변동 중 약 29%를 독립 변수들이 설명합니다.
Adj. R-squared = 0.291	변수 수를 고려한 조정 R <sup>2</sup> 도 비슷하므로 모델은 과도하게 복잡하지 않습니다.
F-statistic = 243.3 (p < 0.001)	전체 회귀 모델이 통계적으로 유의미합니다. (모든 계수가 0이라는 귀무가설 기각)
No. Observations = 6497	데이터 수가 많아 신뢰할 수 있는 결과입니다.

# 1. 데이터 시각화 이해

## ■ 데이터 모델링

### 2. t-검정과 회귀 분석으로 그룹 비교하기(전체 모델 요약)- 개별 변수 해석 (유의미한 변수 중심)

변수	계수 (coef)	p-value	해석
alcohol	+0.267	0.000	알코올 함량이 높을수록 품질 점수가 상승합니다 (가장 영향력 큰 양의 변수 중 하나).
volatile_acidity	-1.3279	0.000	휘발성 산도가 높을수록 품질은 낮아집니다 (강한 음의 관계).
sulphates	+0.7683	0.000	설페이트 함량이 높을수록 품질이 올라갑니다.
density	-54.9669	0.000	밀도가 높을수록 품질은 낮아집니다. (단위가 작아서 계수는 커 보이지만 실제 변화량은 작음)
residual_sugar	+0.0436	0.000	잔당량이 높을수록 품질이 조금 상승함.
free_sulfur_dioxide	+0.0060	0.000	약한 양의 상관관계.
total_sulfur_dioxide	-0.0025	0.000	전체 이산화황이 많을수록 품질은 떨어짐.



# 1. 데이터 시각화 이해

## ■ 데이터 모델링

2. t-검정과 회귀 분석으로 그룹 비교하기(전체 모델 요약)-유의미하지 않은 변수들( $p > 0.05$ )

변수	p-value	해석
citric_acid	0.168	품질과 통계적으로 유의한 관계가 없음.
chlorides	0.146	염소화물 역시 유의하지 않음.

3. 결론 요약

- 가장 긍정적인 영향: alcohol, sulphates, residual\_sugar, pH
- 가장 부정적인 영향: volatile\_acidity, density, total\_sulfur\_dioxide
- 전체 설명력은 29%로, 일부 중요한 요인 외에 감각적 요소나 생산 조건 등이 품질에 영향을 미칠 수 있음.

# 1. 데이터 시각화 이해

## ■ 데이터 모델링

### 3. 회귀 분석 모델로 새로운 샘플의 품질 등급 예측하기

```
sample1 = wine[wine.columns.difference(['quality', 'type'])]
sample1 = sample1[0:5][:]
sample1_predict = regression_result.predict(sample1)
sample1_predict
0    4.997607
1    4.924993
2    5.034663
3    5.680333
4    4.997607
dtype: float64
wine[0:5]['quality']
0    5
1    5
2    5
3    6
4    5
Name: quality, dtype: int64
```

```
data = {"fixed_acidity" : [8.5, 8.1], "volatile_acidity":[0.8, 0.5],
"citric_acid":[0.3, 0.4], "residual_sugar":[6.1, 5.8], "chlorides":[0.055,
0.04], "free_sulfur_dioxide":[30.0, 31.0], "total_sulfur_dioxide":[98.0,
99], "density":[0.996, 0.91], "pH":[3.25, 3.01], "sulphates":[0.4, 0.35],
"alcohol":[9.0, 0.88]}
```

```
sample2 = pd.DataFrame(data, columns= sample1.columns)
sample2
```

	alcohol	chlorides ...	total_sulfur_dioxide	volatile_acidity
0	9.00	0.055 ...	98.0	0.8
1	0.88	0.040 ...	99.0	0.5

[2 rows x 11 columns]

```
sample2_predict = regression_result.predict(sample2)
sample2_predict
0    4.809094
1    7.582129
dtype: float64
```

# 1. 데이터 시각화 이해

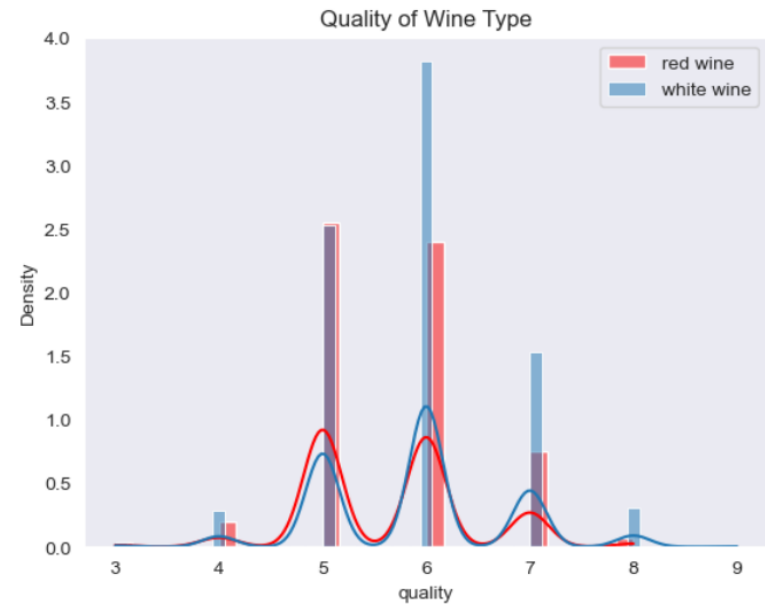
## ■ 결과 시각화

### 1. 와인 유형에 따른 품질 등급 히스토그램 그리기

1) 명령 프롬프트 창에서 다음 명령을 입력하여 seaborn 라이브러리 패키지를 설치

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('dark')

sns.histplot(red_wine_quality, stat='density', kde = True,
             color = "red", label = 'red wine')
sns.histplot(white_wine_quality, stat='density', kde = True,
             label = 'white wine')
plt.title("Quality of Wine Type")
plt.legend()
plt.show()
```



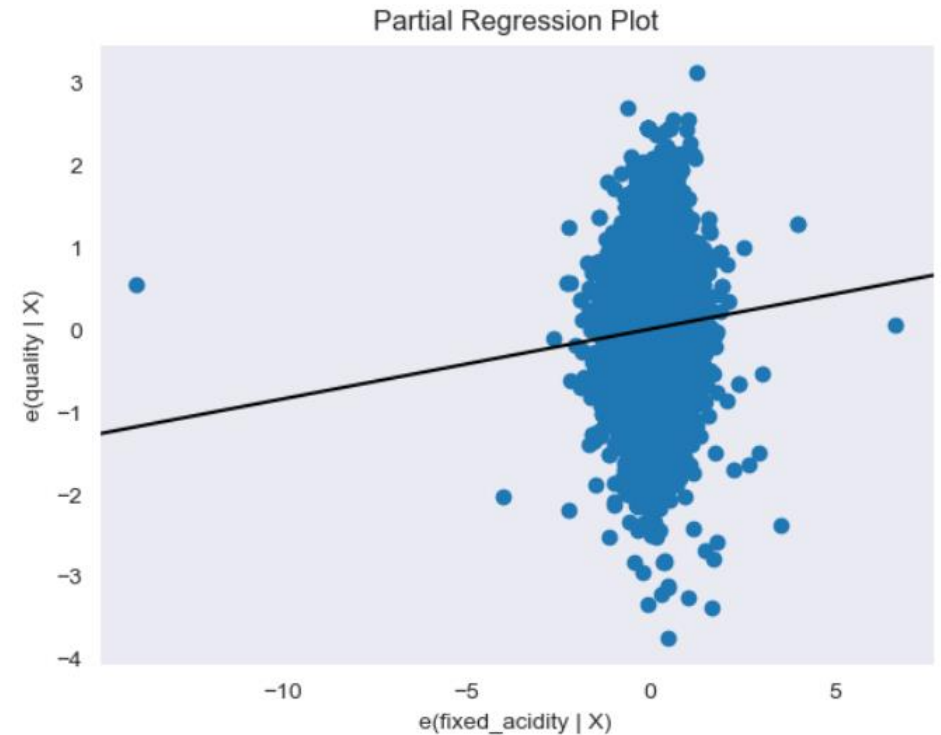
# 1. 데이터 시각화 이해

## ■ 결과 시각화

### 2. 부분 회귀 플롯 시각화

#7.2 부분 회귀 플롯으로 시각화하기

```
import statsmodels.api as sm
others = list(set(wine.columns).difference(set(["quality",
"fixed_acidity"])))
p, resid = sm.graphics.plot_partregress("quality", "fixed_acidity",
others, obs_labels=False, data = wine, ret_coords=True)
plt.show()
```

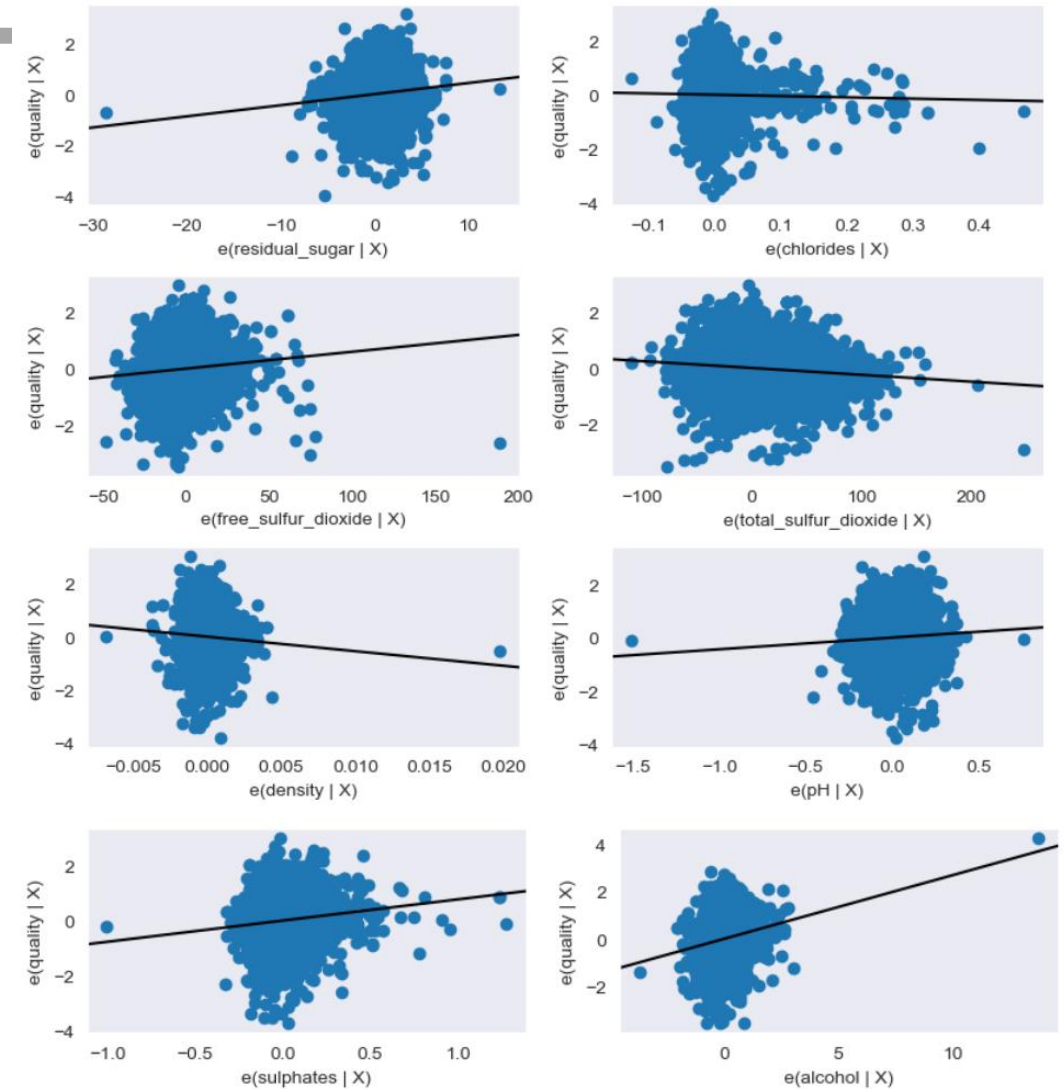
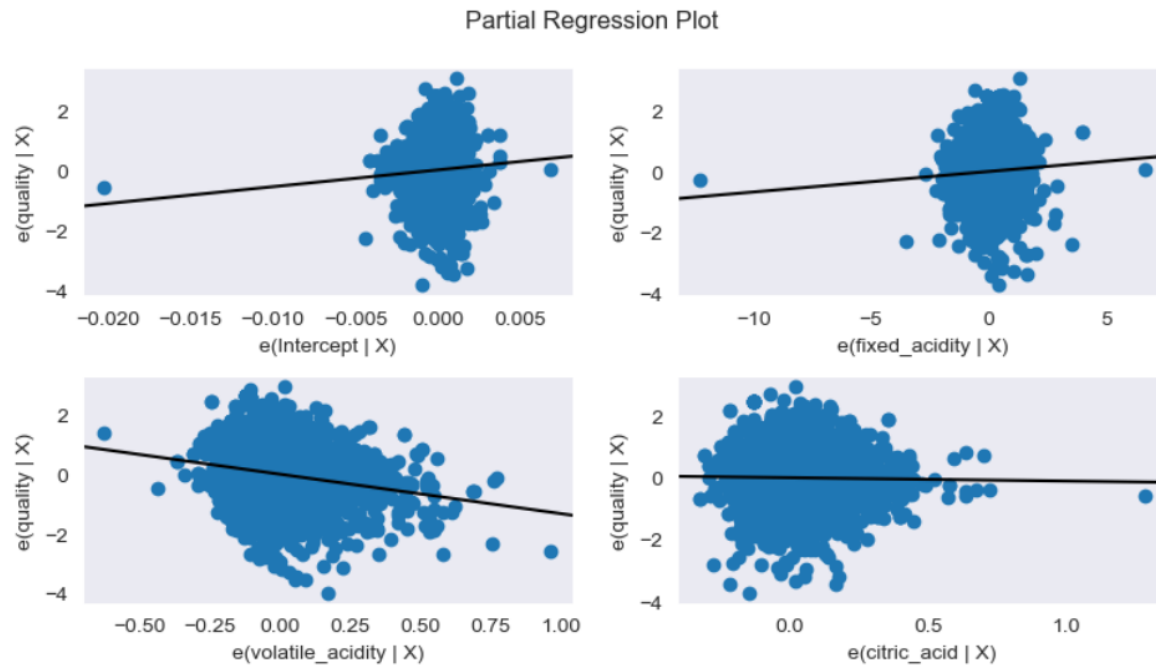


# 1. 데이터 시각화 이해

## ■ 결과 시각화

### 2. 부분 회귀 플롯 시각화

```
fig = plt.figure(figsize = (8, 13))
sm.graphics.plot_partregress_grid(regression_result, fig = fig)
plt.show()
```

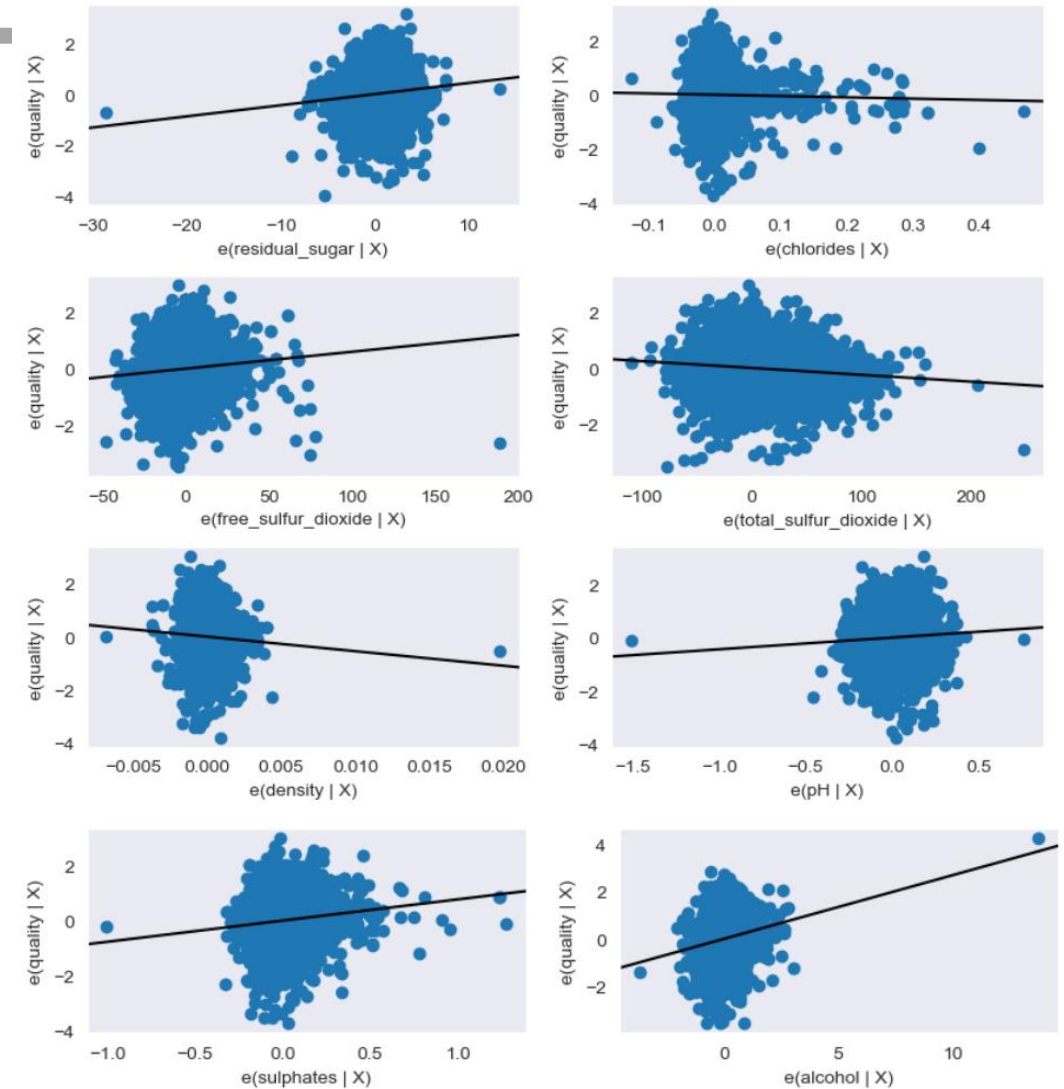
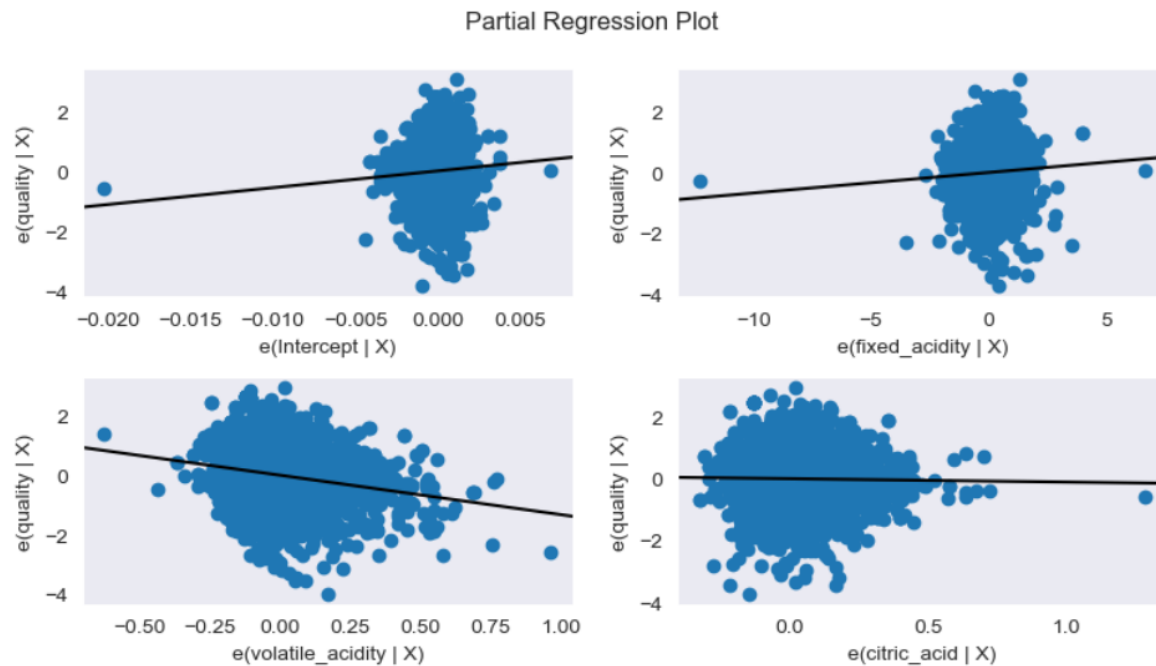


# 1. 데이터 시각화 이해

## ■ 결과 시각화

### 2. 부분 회귀 플롯 시각화

```
fig = plt.figure(figsize = (8, 13))
sm.graphics.plot_partregress_grid(regression_result, fig = fig)
plt.show()
```



02

## 상관관계 분석과 시각화

## 2. 상관관계 분석과 시각화

### ■ 상관 분석(Correlation Analysis)

- 두 변수 간의 선형적 관계의 강도와 방향을 분석하는 통계 기법
- 상관분석의 주요 목적은 한 변수의 변화가 다른 변수와 얼마나 관련되어 있는지를 파악하는 것
- 두 변수는 서로 독립적이거나 상관된 관계일 수 있는데, 두 변수의 관계의 강도를 상관관계 라고함
- 상관 분석에서는 상관관계의 정도를 나타내는 단위로 모상관 계수  $\rho$ 를 사용

### ■ 상관계수(Correlation Coefficient)

- 표기: 일반적으로  $rr$  로 표시됨 (피어슨 상관계수 기준)
- 값의 범위:  $-1 \sim 1$  사이
  - $r=1$  : 완전한 양의 선형 관계
  - $r=-1$  : 완전한 음의 선형 관계
  - $r=0$  : 선형 관계 없음



## 2. 상관관계 분석과 시각화

### ■ 주요 상관계수 종류

종류	설명
피어슨 상관계수 (Pearson's $r$ )	연속형 변수 간의 선형 상관관계 측정
스피어만 상관계수 (Spearman's $\rho$ )	순위형 변수 또는 비선형 관계에서도 사용 가능
켄달의 타우 (Kendall's $\tau$ )	순서쌍의 일관성을 기반으로 측정하는 비모수적 방법

### ■ 해석 예시 (피어슨 상관계수 기준)

상관계수 $r$	상관 정도
0.9 ~ 1.0 or -0.9 ~ -1.0	매우 강한 상관
0.7 ~ 0.9 or -0.7 ~ -0.9	강한 상관
0.4 ~ 0.7 or -0.4 ~ -0.7	중간 정도 상관
0.2 ~ 0.4 or -0.2 ~ -0.4	약한 상관
0.0 ~ 0.2 or -0.0 ~ -0.2	매우 약하거나 없음

- 피어슨 상관 계수
  - 상관 계수 중에서 많이 사용하는 것은 피어슨 상관 계수 Pearson correlation coefficient 또는 Pearson's  $r$
  - 피어슨 상관 계수는  $r$ 로 표현
- 상관 분석 결과의 시각화
  - 상관 분석 결과를 시각화할 때는 두 변수의 관계를 보여주는 산점도나 히트맵을 많이 사용

## 2. 상관관계 분석과 시각화

■ 예)

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# 예시 데이터프레임 생성
data = pd.DataFrame({
    '공부시간': [1, 2, 3, 4, 5],
    '점수': [50, 55, 65, 70, 80]
})

# 상관계수 계산
correlation = data.corr(method='pearson')
print(correlation)

# 상관관계 시각화
sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.show()
```

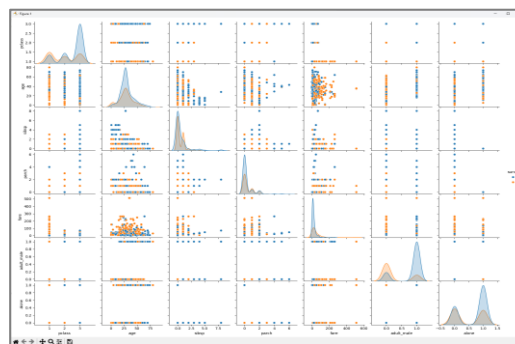
## 2. 상관관계 분석과 시각화

### ■ 분석 미리보기

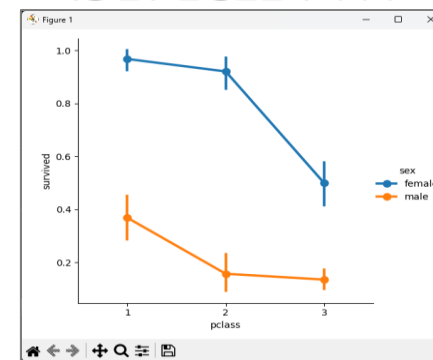
타이타닉호 생존을 분석하기	
목표	타이타닉호 승객 변수를 분석하여 생존율과의 상관관계를 찾는다.
핵심 개념	상관 분석, 상관 계수, 피어슨 상관 계수, 히트맵
데이터 수집	타이타닉 데이터: seaborn 내장 데이터셋
데이터 준비	결측치 치환: 중앙값 치환, 최빈값 치환
데이터 탐색	1. 정보 확인: info() 2. 차트를 통한 데이터 탐색: pie(), countplot()
데이터 모델링	1. 모든 변수 간 상관 계수 구하기 2. 지정한 두 변수 간 상관관계 구하기

### 결과 시각화

#### 1. 산점도를 이용한 시각화



#### 2. 특정 변수 간 상관관계 시각화



#### 3. 히트맵을 이용한 시각화



## 2. 상관관계 분석과 시각화

### ■ 목표 설정

- 타이타닉호의 생존자와 관련된 변수의 상관관계 찾아보기
- 생존과 가장 상관도가 높은 변수는 무엇인지 분석
- 상관 분석을 위해 피어슨 상관 계수를 사용
- 변수 간의 상관관계는 시각화하여 분석

## 2. 상관관계 분석과 시각화

### ■ 데이터 수집

#### #3 데이터 수집

```
import seaborn as sns
import pandas as pd
titanic = sns.load_dataset("titanic")
titanic.to_csv('data/titanic.csv', index = False)
```

### ■ 데이터 정리

- 저장한 titanic.csv 파일을 열어서 데이터 정리 작업이 필요한지 확인하기

파일

홈

삽입

페이지 레이아웃

수식

데이터

검토

보기

부하 테스트

팀

A1

f\_x

survived

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_t	alive	alone
2	0	3	male	22	1	0	7.25	S	Third	man	TRUE		Southamp	no	FALSE
3	1	1	female	38	1	0	71.2833	C	First	woman	FALSE	C	Cherbourg	yes	FALSE
4	1	3	female	26	0	0	7.925	S	Third	woman	FALSE		Southamp	yes	TRUE
5	1	1	female	35	1	0	53.1	S	First	woman	FALSE	C	Southamp	yes	FALSE
6	0	3	male	35	0	0	8.05	S	Third	man	TRUE		Southamp	no	TRUE
7	0	3	male		0	0	8.4583	Q	Third	man	TRUE		Queensto	no	TRUE

titanic

준비

## 2. 상관관계 분석과 시각화

### ■ 데이터 준비

```
titanic.isnull().sum()
titanic['age'] = titanic['age'].fillna(titanic['age'].median())
titanic['embarked'].value_counts()
titanic['embarked'] = titanic['embarked'].fillna('S')
titanic['embark_town'].value_counts()
titanic['embark_town'] = titanic['embark_town'].fillna('Southampton')
titanic['deck'].value_counts()
titanic['deck'] = titanic['deck'].fillna('C')
titanic.isnull().sum()
```

## 2. 상관관계 분석과 시각화

### ■ 데이터 탐색

#### ■ 데이터의 기본 정보 탐색하기

```
titanic.info()
```

```
titanic.survived.value_counts()
```

```
01 >>> titanic.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   survived    891 non-null    int64
1   pclass      891 non-null    int64
2   sex         891 non-null    object
2   sex         891 non-null    object
3   age         891 non-null    float64
4   sibsp       891 non-null    int64
5   parch       891 non-null    int64
6   fare        891 non-null    float64
7   embarked    891 non-null    object
8   class       891 non-null    category
9   who         891 non-null    object
10  adult_male  891 non-null    bool
11  deck        891 non-null    category
12  embark_town 891 non-null    object
13  alive       891 non-null    object
14  alone       891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.6+ KB

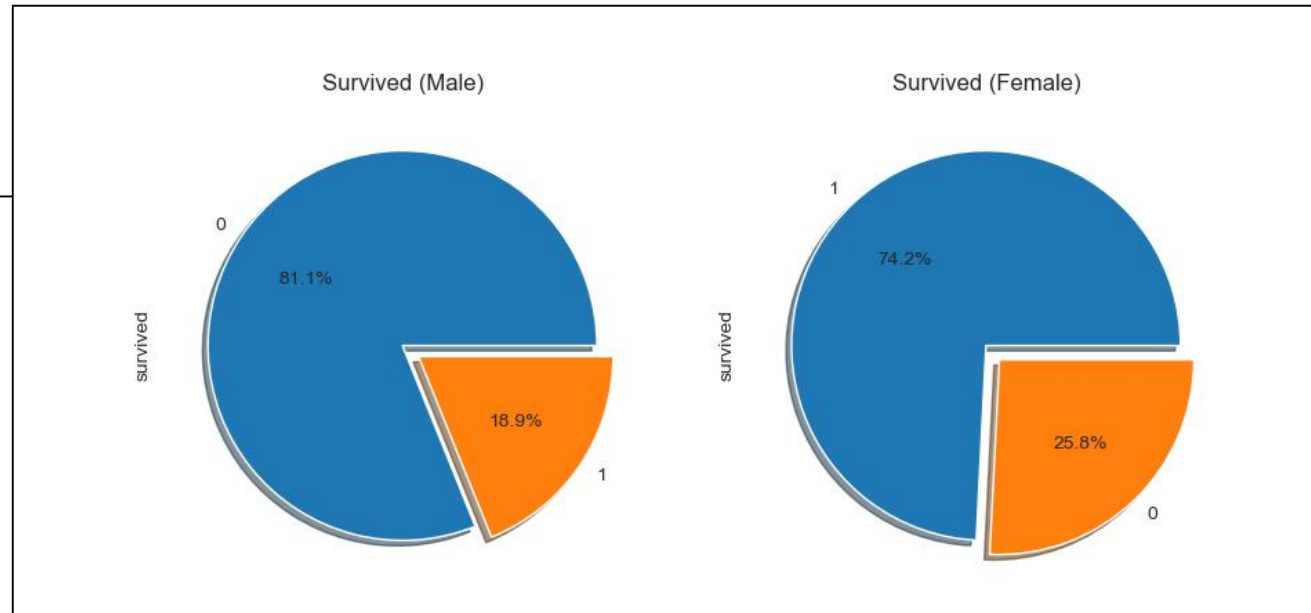
02 >>> titanic.survived.value_counts()
0    549
1    342
Name: survived, dtype: int64
```

## 2. 상관관계 분석과 시각화

### ■ 데이터 탐색

- 차트를 그려서 데이터를 시각적으로 탐색하기

```
import matplotlib.pyplot as plt
f,ax = plt.subplots(1, 2, figsize = (10, 5))
titanic['survived'][titanic['sex'] == 'male'].value_counts().plot.pie(explode = [0,0.1], autopct = '%1.1f%%', ax = ax[0],
shadow = True)
titanic['survived'][titanic['sex'] == 'female'].value_counts().plot.pie(explode = [0,0.1], autopct = '%1.1f%%', ax = ax[1],
shadow = True)
ax[0].set_title('Survived (Male)')
ax[1].set_title('Survived (Female)')
plt.show()
```



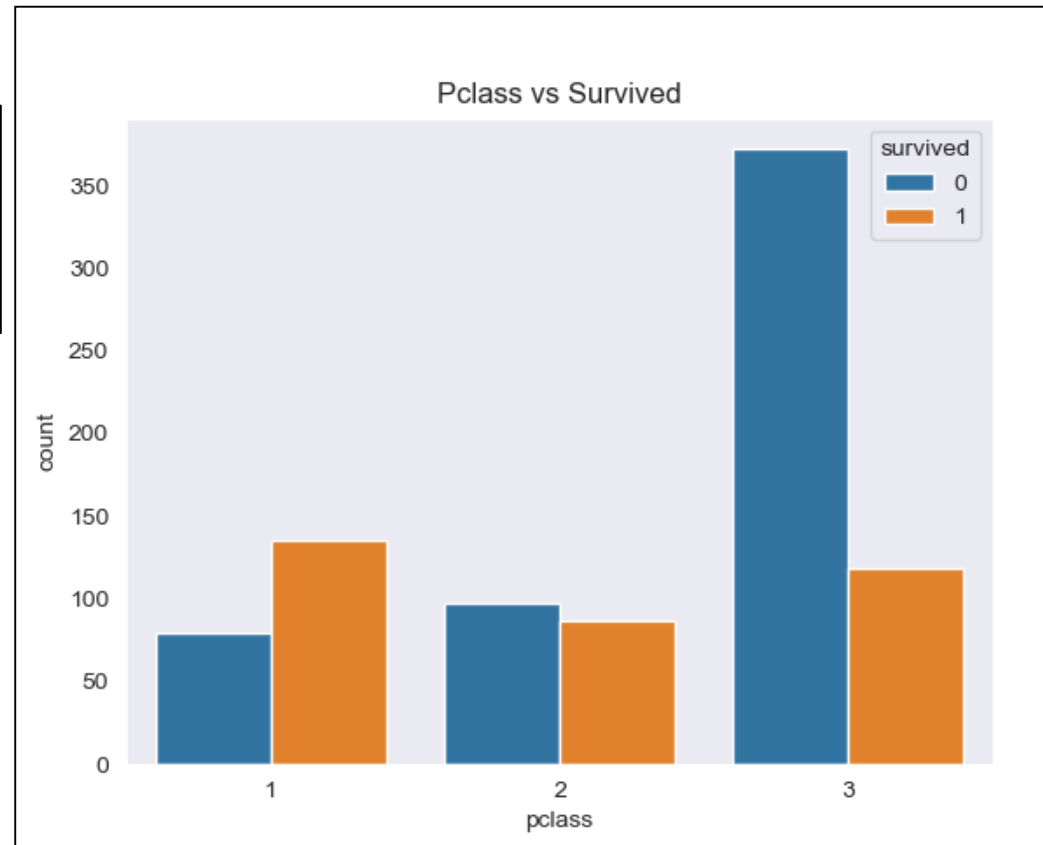


## 2. 상관관계 분석과 시각화

### ■ 데이터 탐색

- 등급별 생존자 수를 차트로 나타내기

```
sns.countplot(x = 'pclass', hue = 'survived', data = titanic)  
plt.title('Pclass vs Survived')  
plt.show()
```



## 2. 상관관계 분석과 시각화

### ■ 데이터 모델링

- 상관관계 분석을 위한 상관계수 구하고 저장하기

```
titanic2 = titanic.select_dtypes(include=[int, float, bool])
titanic2.shape
titanic_corr = titanic2.corr(method = 'pearson')
titanic_corr
titanic_corr.to_csv('data/titanic_corr.csv', index = False)
```

	survived	pclass	age	...	fare	adult_male	alone
survived	1.000000	-0.338481	-0.064910	...	0.257307	-0.557080	-0.203367
pclass	-0.338481	1.000000	-0.339898	...	-0.549500	0.094035	0.135207
age	-0.064910	-0.339898	1.000000	...	0.096688	0.247704	0.171647
sibsp	-0.035322	0.083081	-0.233296	...	0.159651	-0.253586	-0.584471
parch	0.081629	0.018443	-0.172482	...	0.216225	-0.349943	-0.583398
fare	0.257307	-0.549500	0.096688	...	1.000000	-0.182024	-0.271832
adult_male	-0.557080	0.094035	0.247704	...	-0.182024	1.000000	0.404744
alone	-0.203367	0.135207	0.171647	...	-0.271832	0.404744	1.000000

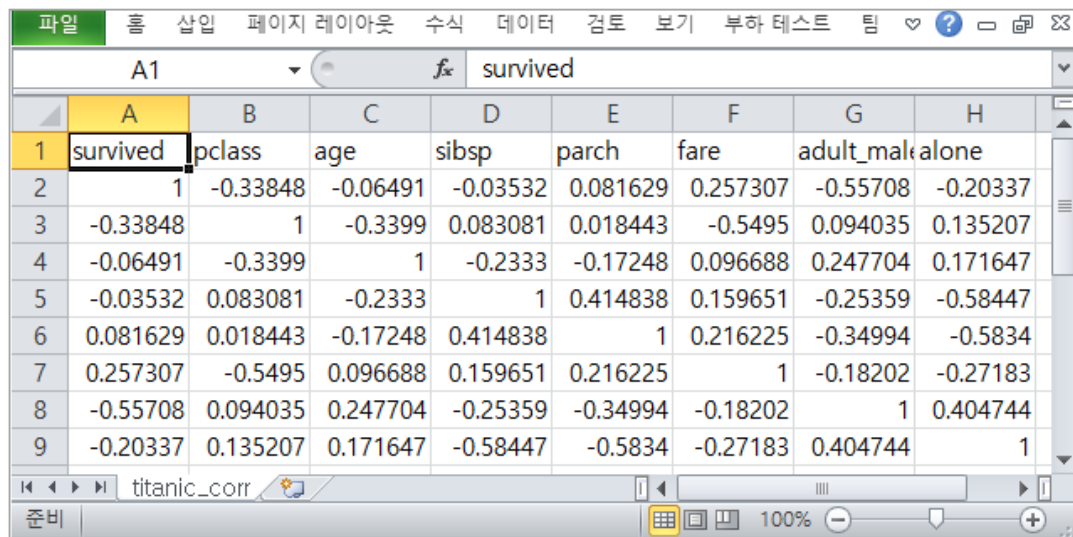
[8 rows x 8 columns]

## 2. 상관관계 분석과 시각화

### ■ 데이터 모델링

#### ■ 상관 계수 확인하기

- 남자 성인(adult\_male): 생존(survived)과 음의 상관관계
- 객실 등급(pclass): 음의 상관관계
- 관계, 객실 요금fare은 양의 상관관계
- 동행 없이 혼자 탑승한 경우(alone): 생존율이 떨어진다는 상관관계가 확인됨



	A	B	C	D	E	F	G	H
1	survived	pclass	age	sibsp	parch	fare	adult_male	alone
2	1	-0.33848	-0.06491	-0.03532	0.081629	0.257307	-0.55708	-0.20337
3	-0.33848	1	-0.3399	0.083081	0.018443	-0.5495	0.094035	0.135207
4	-0.06491	-0.3399	1	-0.2333	-0.17248	0.096688	0.247704	0.171647
5	-0.03532	0.083081	-0.2333	1	0.414838	0.159651	-0.25359	-0.58447
6	0.081629	0.018443	-0.17248	0.414838	1	0.216225	-0.34994	-0.5834
7	0.257307	-0.5495	0.096688	0.159651	0.216225	1	-0.18202	-0.27183
8	-0.55708	0.094035	0.247704	-0.25359	-0.34994	-0.18202	1	0.404744
9	-0.20337	0.135207	0.171647	-0.58447	-0.5834	-0.27183	0.404744	1

## 2. 상관관계 분석과 시각화

### ■ 데이터 모델링

- 특정 변수 사이의 상관관계 구하기

```
titanic['survived'].corr(titanic['adult_male'])  
titanic['survived'].corr(titanic['fare'])
```

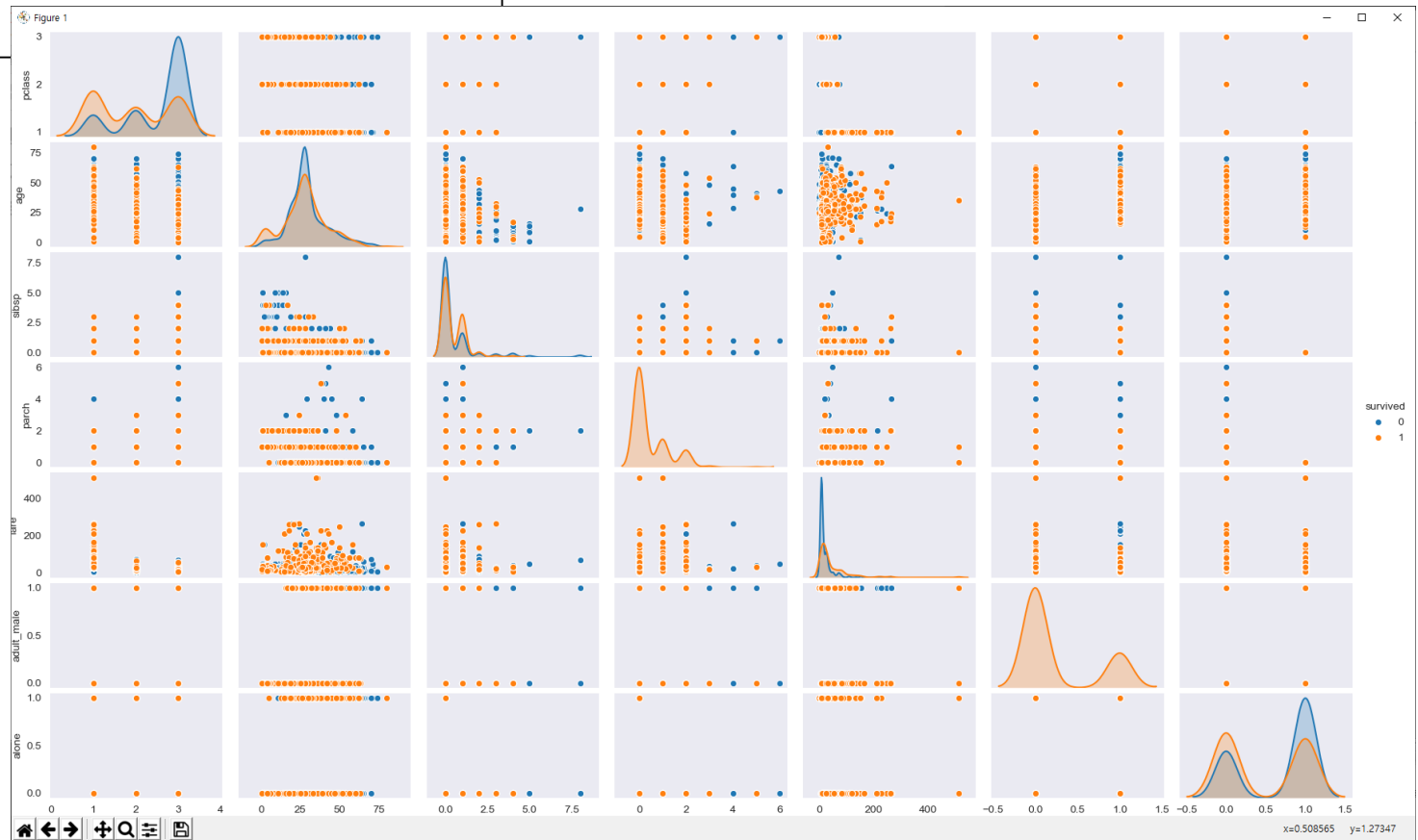
```
titanic['survived'].corr(titanic['adult_male'])  
-0.5570800422053259  
titanic['survived'].corr(titanic['fare'])  
0.2573065223849622
```

## 2. 상관관계 분석과 시각화

### ■ 결과 시각화

- 삼점도로 상관 분석 시각화하기

```
sns.pairplot(titanic, hue = 'survived')  
plt.show()
```

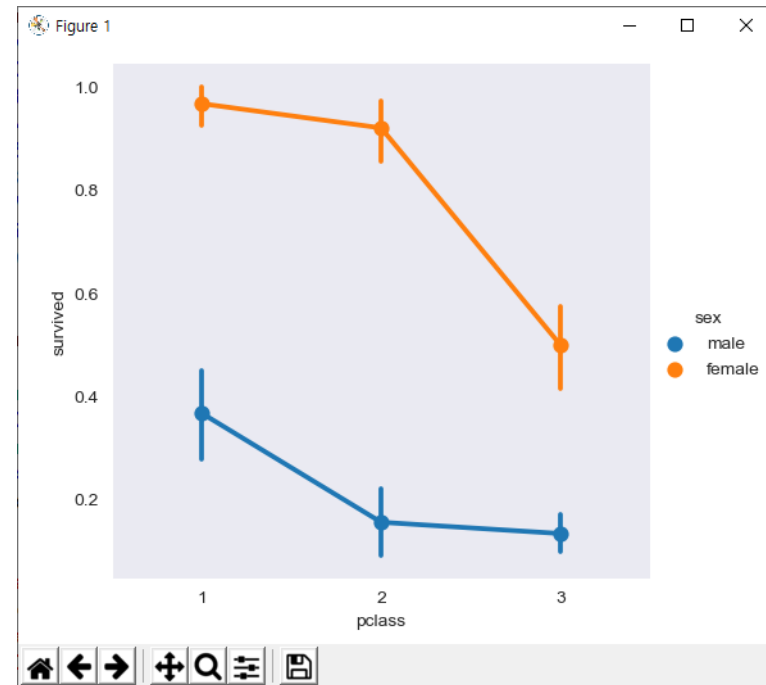


## 2. 상관관계 분석과 시각화

### ■ 결과 시각화

- 두 변수의 상관관계 시각화하기
  - 생존자와 객실 등급, 성별관계를 catplot() 시각화

```
sns.catplot(x = 'pclass', y = 'survived',  
            hue = 'sex', data = titanic, kind = 'point')  
plt.show()
```



## 2. 상관관계 분석과 시각화

### ■ 결과 시각화

- 변수 사이의 상관계수를 히트맵으로 시각화하기

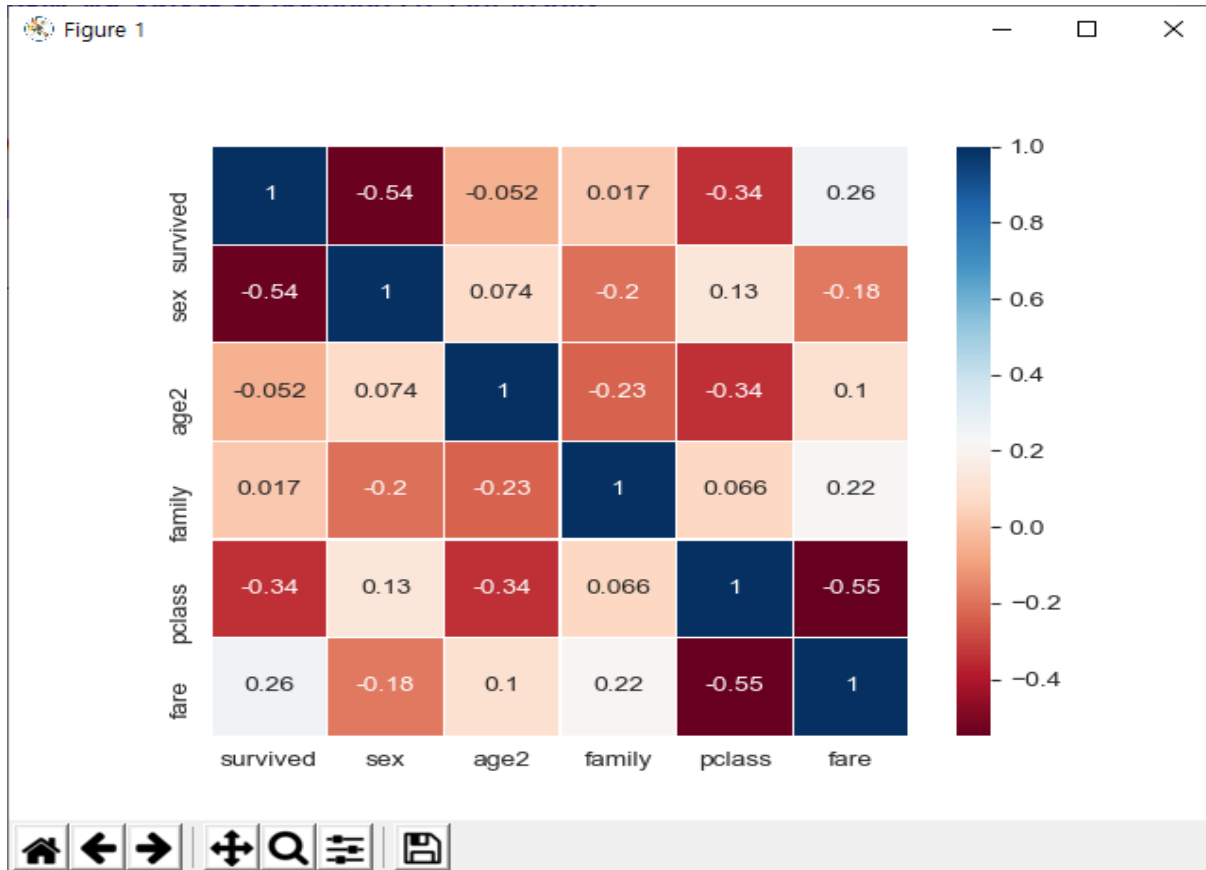
```
def category_age(x):  
    if x < 10:  
        return 0  
    elif x < 20:  
        return 1  
    elif x < 30:  
        return 2  
    elif x < 40:  
        return 3  
    elif x < 50:  
        return 4  
    elif x < 60:  
        return 5  
    elif x < 70:  
        return 6  
    else:  
        return 7
```

```
titanic['age2'] = titanic['age'].apply(category_age)  
titanic['sex'] = titanic['sex'].map({'male':1, 'female':0})  
titanic['family'] = titanic['sibsp'] + titanic['parch'] + 1  
titanic.to_csv('data/titanic3.csv', index = False)  
heatmap_data = titanic[['survived', 'sex', 'age2', 'family', 'pclass', 'fare']]  
colormap = plt.cm.RdBu  
sns.heatmap(heatmap_data.astype(float).corr(), linewidths = 0.1,  
            vmax = 1.0, square = True, cmap = colormap, linecolor = 'white',  
            annot = True, annot_kws = {"size": 10})  
  
plt.show()
```

## 2. 상관관계 분석과 시각화

### ■ 결과 시각화

- 변수 사이의 상관계수를 히트맵으로 시각화하기





03

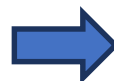
통계분석 실습

### 3. 통계분석 실습

#### ■ 분석 미리보기

- 삶의 만족도에 대한 건강, 경제, 관계 및 사회참여, 교육, 안전, 여가, 환경 데이터가 삶의 만족도에 미치는 영향을 회귀 분석과 상관관계 분석을 수행하고 시각화를 수행
- 회귀분석 결과 요약

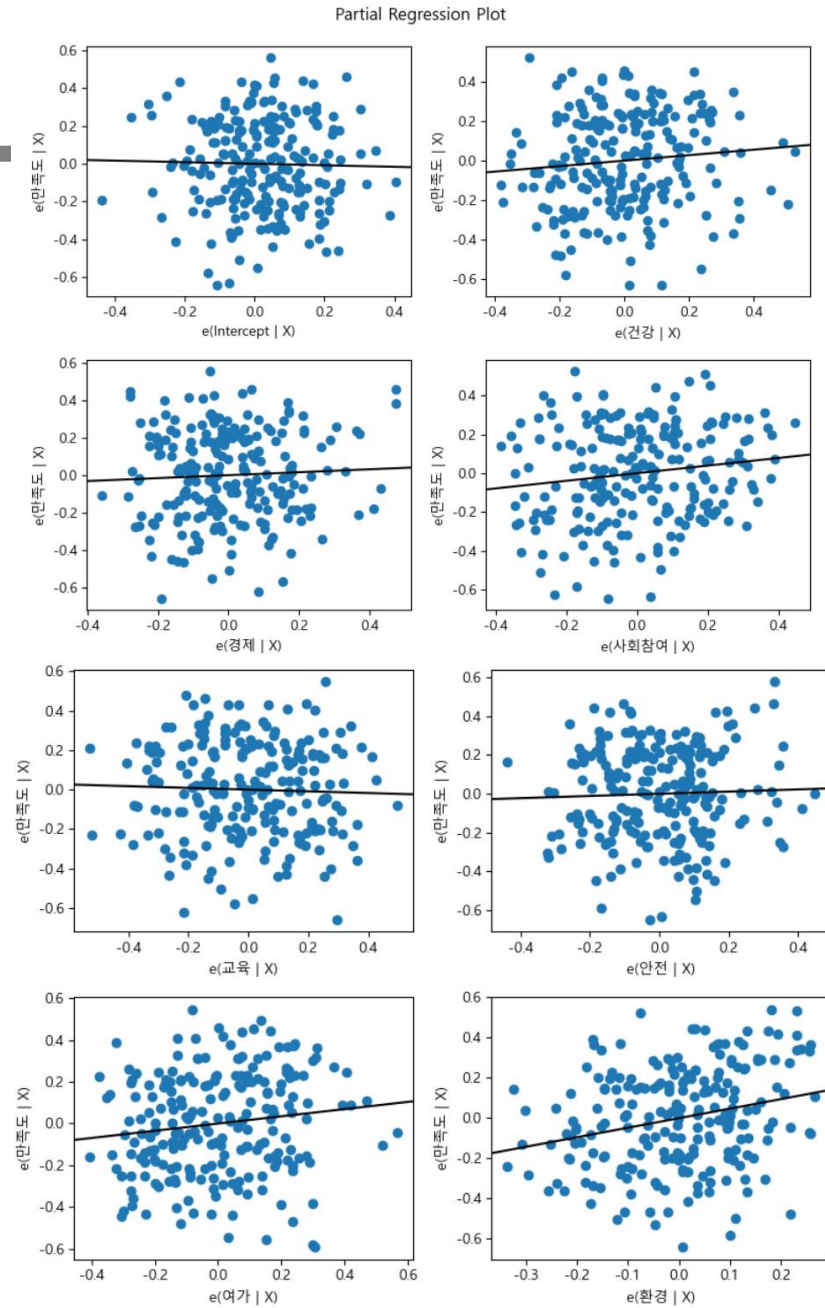
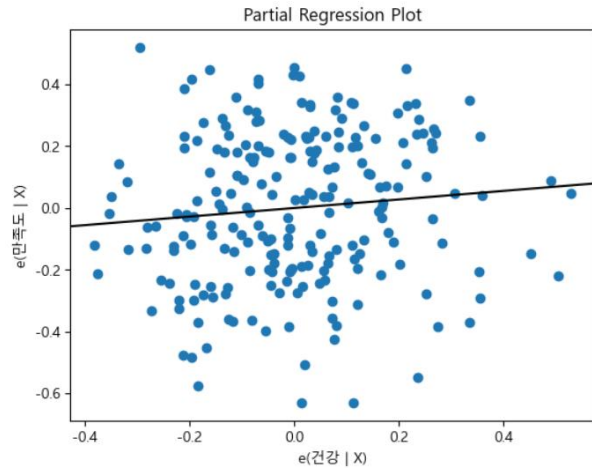
OLS Regression Results			
Dep. Variable:	만족도	R-squared:	0.184
Model:	OLS	Adj. R-squared:	0.158
Method:	Least Squares	F-statistic:	6.941
Date:	Tue, 20 May 2025	Prob (F-statistic):	1.87e-07
Time:	12:16:20	Log-Likelihood:	1.1660
No. Observations:	223	AIC:	13.67
Df Residuals:	215	BIC:	40.93
Df Model:	7		
Covariance Type:	nonrobust		



	coef	std err	t	P> t	[0.025	0.975]
Intercept	-0.0407	0.118	-0.343	0.732	-0.274	0.193
건강	0.1385	0.095	1.464	0.145	-0.048	0.325
경제	0.0789	0.106	0.747	0.456	-0.129	0.287
사회참여	0.1961	0.090	2.178	0.030	0.019	0.374
교육	-0.0434	0.082	-0.529	0.598	-0.205	0.118
안전	0.0564	0.104	0.540	0.590	-0.149	0.262
여가	0.1731	0.086	2.012	0.045	0.004	0.343
환경	0.4746	0.126	3.775	0.000	0.227	0.722
Omnibus:	5.538	Durbin-Watson:	2.205			
Prob(Omnibus):	0.063	Jarque-Bera (JB):	3.994			
Skew:	-0.192	Prob(JB):	0.136			
Kurtosis:	2.469	Cond. No.	17.3			

### 3. 통계분석 실습

- 분석 미리보기
  - 회귀분석 결과 시각화

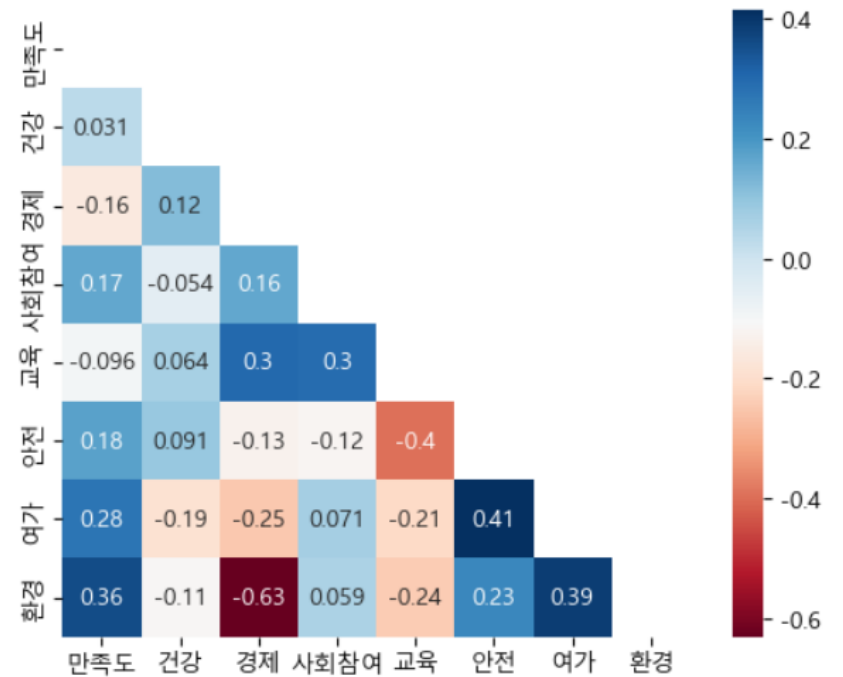


### 3. 통계분석 실습

#### ■ 분석결과 미리보기

##### ■ 상관관계 분석 및 결과 시각화

	만족도	건강	경제	사회참여	교육	안전	여가	환경
만족도	1.000000	0.030889	-0.159564	0.165182	-0.095661	0.175283	0.275589	0.357748
건강	0.030889	1.000000	0.118797	-0.054158	0.063818	0.090740	-0.186737	-0.112172
경제	-0.159564	0.118797	1.000000	0.163335	0.302246	-0.129147	-0.247433	-0.630315
사회참여	0.165182	-0.054158	0.163335	1.000000	0.295712	-0.118827	0.070937	0.059245
교육	-0.095661	0.063818	0.302246	0.295712	1.000000	-0.396170	-0.213909	-0.241198
안전	0.175283	0.090740	-0.129147	-0.118827	-0.396170	1.000000	0.414465	0.233188
여가	0.275589	-0.186737	-0.247433	0.070937	-0.213909	0.414465	1.000000	0.386590
환경	0.357748	-0.112172	-0.630315	0.059245	-0.241198	0.233188	0.386590	1.000000



### 3. 데이터 분석 및 시각화

#### ■ 데이터 준비

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from statsmodels.formula.api import ols, glm
```

```
happy_만족도=pd.read_excel('data4/삶의만족도.xlsx')
happy_만족도.head()
happy_건강=pd.read_excel('data4/건강.xlsx')
happy_건강.head()
# happy_경제, happy_사회참여, happy_교육, happy_안전, happy_여가, happy_환경을
같은 방법으로 파일을 로드하여 DataFrame을 생성한다.
```

### 3. 통계분석 실습

#### ■ 데이터 준비

- 아래의 코드를 참고하여 최종 DataFrame과 작성하라.

```
happy_df=happy_만족도
happy_df['건강']=happy_건강['평균']
happy_df['경제']=happy_경제['평균']
...
```

No	시도	구군	삶의 만족도	건강	경제	사회참여	교육	안전	여가	환경
0	1	서울특별시 종로구	0.4437	0.9220	1.0000	0.7425	0.6839	0.7470	0.6331	0.4637
1	2	서울특별시 중구	0.4976	0.6742	0.9806	0.4608	0.5013	0.9320	0.6691	0.2865
2	3	서울특별시 용산구	0.6161	0.5898	0.6915	0.4317	0.2679	0.5537	0.2817	0.5030
3	4	서울특별시 성동구	0.4729	0.4794	0.6533	0.4182	0.2464	0.5347	0.3257	0.4196
4	5	서울특별시 광진구	0.4041	0.6373	0.4445	0.3519	0.4879	0.6072	0.3313	0.4992

### 3. 통계분석 실습

#### ■ 데이터 결측치 제거

- happy\_df에 null 값이 있는지 검사한 결과 다음과 같이 나타난다.

```
No          0
시도        0
구군        0
삶의 만족도  1
건강        1
경제        1
사회참여    1
교육        1
안전        1
여가        1
환경        1
dtype: int64
```

```
# 삶의 만족도의 결측치를 삶의 만족도 필드의 평균 값으로 채워 결측치 처리
happy_df1=
```

```
# 건강, 경제 등의 필드의 결측치 처리는 null 값을 포함한 행을 제거한다.
happy_dh1=
```

### 3. 통계분석 실습

#### ■ 데이터 탐색

- happy\_df와 happy\_df1의 데이터 정보 보기

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 229 entries, 0 to 228
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   No          229 non-null   int64  
 1   시도        229 non-null   object  
 2   구군        229 non-null   object  
 3   삶의 만족도  229 non-null   float64 
 4   건강        228 non-null   float64 
 5   경제        228 non-null   float64 
 6   사회참여    228 non-null   float64 
 7   교육        228 non-null   float64 
 8   안전        228 non-null   float64 
 9   여가        228 non-null   float64 
10  환경        228 non-null   float64 
dtypes: float64(8), int64(1), object(2)
memory usage: 19.8+ KB
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 223 entries, 0 to 228
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   No          223 non-null   int64  
 1   시도        223 non-null   object  
 2   구군        223 non-null   object  
 3   삶의 만족도  223 non-null   float64 
 4   건강        223 non-null   float64 
 5   경제        223 non-null   float64 
 6   사회참여    223 non-null   float64 
 7   교육        223 non-null   float64 
 8   안전        223 non-null   float64 
 9   여가        223 non-null   float64 
10  환경        223 non-null   float64 
dtypes: float64(8), int64(1), object(2)
memory usage: 20.9+ KB
```



### 3. 통계분석 실습

#### ■ 데이터 분석

- 다음과 같이 통계요약(describe) 구하라

	No	삶의 만족도	건강	경제	사회참여	교육	안전	여가	환경
count	223.000000	223.000000	223.000000	223.000000	223.000000	223.000000	223.000000	223.000000	223.000000
mean	114.950673	0.495216	0.405873	0.392500	0.470994	0.542274	0.455517	0.463204	0.577374
std	65.925239	0.267121	0.181590	0.211245	0.199592	0.235800	0.190406	0.229254	0.181753
min	1.000000	0.005200	0.005500	0.008600	0.009400	0.013900	0.031100	0.023800	0.073300
25%	59.500000	0.292850	0.268300	0.230850	0.339250	0.370550	0.313850	0.278200	0.455200
50%	115.000000	0.487900	0.389700	0.378800	0.473200	0.563200	0.446100	0.447800	0.598700
75%	170.500000	0.699550	0.525100	0.513150	0.617850	0.730250	0.602500	0.647400	0.701300
max	229.000000	0.999000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

### 3. 통계분석 실습

#### ■ 데이터 탐색

- happy\_df1을 사용하여 다음과 같이 시도의 value\_counts를 구하라

```
 시도
경기도      31
서울특별시  25
경상북도    23
전라남도    22
강원도      18
경상남도    18
부산광역시  16
충청남도    15
전라북도    14
충청북도    11
인천광역시  10
대구광역시   8
광주광역시   5
대전광역시   5
울산광역시   5
제주특별자치도  2
세종특별자치시  1
Name: count, dtype: int64
```

### 3. 통계분석 실습

#### ■ 데이터 분석(groupby)

- happy\_df1을 사용하여 '시도'로 그룹을 하여 '삶의 만족도'의 평균을 구하라.
- happy\_df1을 사용하여 '시도'로 그룹을 하여 ['삶의 만족도', '건강']의 평균과 합을 구하라.

```
시도
강원도      0.619506
경기도      0.426023
경상남도    0.530794
경상북도    0.477836
광주광역시  0.484480
대구광역시  0.432833
대전광역시  0.407580
부산광역시  0.365787
서울특별시  0.490972
세종특별자치시  0.907700
울산광역시  0.471980
인천광역시  0.411480
전라남도    0.549557
전라북도    0.608193
제주특별자치도  0.711300
Name: 삶의 만족도, dtype: float64
```

	삶의 만족도		건강	
	mean	sum	mean	sum
시도				
강원도	0.619506	11.151100	0.329506	5.9311
경기도	0.426023	13.206700	0.353952	10.9725
경상남도	0.530794	9.554300	0.312722	5.6290
경상북도	0.477836	10.034562	0.271990	5.7118
광주광역시	0.484480	2.422400	0.632300	3.1615
대구광역시	0.432833	2.597000	0.538483	3.2309
대전광역시	0.407580	2.037900	0.663580	3.3179
부산광역시	0.365787	5.486800	0.517073	7.7561
서울특별시	0.490972	12.274300	0.569532	14.2383
세종특별자치시	0.907700	0.907700	0.232000	0.2320
울산광역시	0.471980	2.359900	0.420040	2.1002
인천광역시	0.411480	4.114800	0.339620	3.3962
전라남도	0.549557	11.540700	0.419205	8.8033
전라북도	0.608193	8.514700	0.421300	5.8982
제주특별자치도	0.711300	1.422600	0.252700	0.5074

### 3. 통계분석 실습

#### ■ 데이터 분석(groupby)

- happy\_df1을 사용하여 '시도'로 그룹을 하여 '삶의 만족도'는 평균을, '건강'은 합을 구하라.
- happy\_df1을 사용하여 '시도'로 그룹을 하여 '삶의 만족도'는 평균과 중앙값을, '건강'은 합과 표준편차를 구하라.

시도	삶의 만족도		건강
	mean	median	std
강원도	0.619506	0.712350	5.9311
경기도	0.426023	0.352900	10.9725
경상남도	0.530794	0.468400	5.6290
경상북도	0.477836	0.493762	5.7118
광주광역시	0.484480	0.475600	3.1615
대구광역시	0.432833	0.413850	3.2309
대전광역시	0.407580	0.293500	3.3179
부산광역시	0.365787	0.423900	7.7561
서울특별시	0.490972	0.497600	14.2383
세종특별자치시	0.907700	0.907700	0.2320
울산광역시	0.471980	0.476400	2.1002
인천광역시	0.411480	0.340600	3.3962
전라남도	0.549557	0.567900	8.8033
전라북도	0.608193	0.600200	5.8982
제주특별자치도	0.711300	0.711300	0.5074
충청남도	0.553100	0.553100	5.2115

시도	삶의 만족도		건강	
	mean	median	sum	std
강원도	0.619506	0.712350	5.9311	0.096995
경기도	0.426023	0.352900	10.9725	0.143195
경상남도	0.530794	0.468400	5.6290	0.132623
경상북도	0.477836	0.493762	5.7118	0.143078
광주광역시	0.484480	0.475600	3.1615	0.166054
대구광역시	0.432833	0.413850	3.2309	0.303610
대전광역시	0.407580	0.293500	3.3179	0.133533
부산광역시	0.365787	0.423900	7.7561	0.206349
서울특별시	0.490972	0.497600	14.2383	0.147599
세종특별자치시	0.907700	0.907700	0.2320	NaN
울산광역시	0.471980	0.476400	2.1002	0.143179
인천광역시	0.411480	0.340600	3.3962	0.116198
전라남도	0.549557	0.567900	8.8033	0.171739
전라북도	0.608193	0.600200	5.8982	0.094197

### 3. 통계분석 실습

#### ■ 데이터 분석(회귀분석)

- 회귀분석을 위해 아래 그림과 같이 happy\_df1에서 ['삶의 만족도' ~ '환경'] 필드만 선택하라.

```
data=(  
    data
```

	삶의 만족도	건강	경제	사회참여	교육	안전	여가	환경
0	0.4437	0.9220	1.0000	0.7425	0.6839	0.7470	0.6331	0.4637
1	0.4976	0.6742	0.9806	0.4608	0.5013	0.9320	0.6691	0.2865
2	0.6161	0.5898	0.6915	0.4317	0.2679	0.5537	0.2817	0.5030
3	0.4729	0.4794	0.6533	0.4182	0.2464	0.5347	0.3257	0.4196
4	0.4041	0.6373	0.4445	0.3519	0.4879	0.6072	0.3313	0.4992
...	...	...	...	...	...	...	...	...
224	0.9565	0.2036	0.1401	0.2192	0.5064	0.7012	0.6758	0.7717

### 3. 통계분석 실습

#### ■ 데이터 분석(회귀분석)

- data에서 필드명이 '삶의 만족도'를 '만족도'로 변경하라.(rename() 사용)

```
data=(  
data
```

	삶의 만족도	건강	경제	사회참여	교육	안전	여가	환경
0	0.4437	0.9220	1.0000	0.7425	0.6839	0.7470	0.6331	0.4637
1	0.4976	0.6742	0.9806	0.4608	0.5013	0.9320	0.6691	0.2865
2	0.6161	0.5898	0.6915	0.4317	0.2679	0.5537	0.2817	0.5030
3	0.4729	0.4794	0.6533	0.4182	0.2464	0.5347	0.3257	0.4196
4	0.4041	0.6373	0.4445	0.3519	0.4879	0.6072	0.3313	0.4992
...	...	...	...	...	...	...	...	...
224	0.9565	0.2036	0.1401	0.2192	0.5064	0.7012	0.6758	0.7717

	만족도	건강	경제	사회참여	교육	안전	여가	환경
0	0.4437	0.9220	1.0000	0.7425	0.6839	0.7470	0.6331	0.4637
1	0.4976	0.6742	0.9806	0.4608	0.5013	0.9320	0.6691	0.2865
2	0.6161	0.5898	0.6915	0.4317	0.2679	0.5537	0.2817	0.5030
3	0.4729	0.4794	0.6533	0.4182	0.2464	0.5347	0.3257	0.4196
4	0.4041	0.6373	0.4445	0.3519	0.4879	0.6072	0.3313	0.4992
...	...	...	...	...	...	...	...	...
224	0.9565	0.2036	0.1401	0.2192	0.5064	0.7012	0.6758	0.7717

### 3. 통계분석 실습

#### ■ 데이터 분석(회귀분석)

- 종속변수 : 만족도(y), 독립변수(건강, 경제, 사회참여, 교육, 안전, 여가, 환경)를 사용하여 회귀분석 식(Rformula)을 작성하고, ols 메소드를 사용하여 회귀분석을 수행하고, 결과를 표시하라.

```
Rformula=  
regression_result=  
regression_result.summary()
```

#### OLS Regression Results

Dep. Variable:	만족도	R-squared:	0.184
Model:	OLS	Adj. R-squared:	0.158
Method:	Least Squares	F-statistic:	6.941
Date:	Tue, 20 May 2025	Prob (F-statistic):	1.87e-07
Time:	12:16:20	Log-Likelihood:	1.1660
No. Observations:	223	AIC:	13.67
Df Residuals:	215	BIC:	40.93
Df Model:	7		
Covariance Type:	nonrobust		



	coef	std err	t	P> t	[0.025	0.975]
Intercept	-0.0407	0.118	-0.343	0.732	-0.274	0.193
건강	0.1385	0.095	1.464	0.145	-0.048	0.325
경제	0.0789	0.106	0.747	0.456	-0.129	0.287
사회참여	0.1961	0.090	2.178	0.030	0.019	0.374
교육	-0.0434	0.082	-0.529	0.598	-0.205	0.118
안전	0.0564	0.104	0.540	0.590	-0.149	0.262
여가	0.1731	0.086	2.012	0.045	0.004	0.343
환경	0.4746	0.126	3.775	0.000	0.227	0.722
Omnibus:	5.538	Durbin-Watson:	2.205			
Prob(Omnibus):	0.063	Jarque-Bera (JB):	3.994			
Skew:	-0.192	Prob(JB):	0.136			
Kurtosis:	2.469	Cond. No.	17.3			

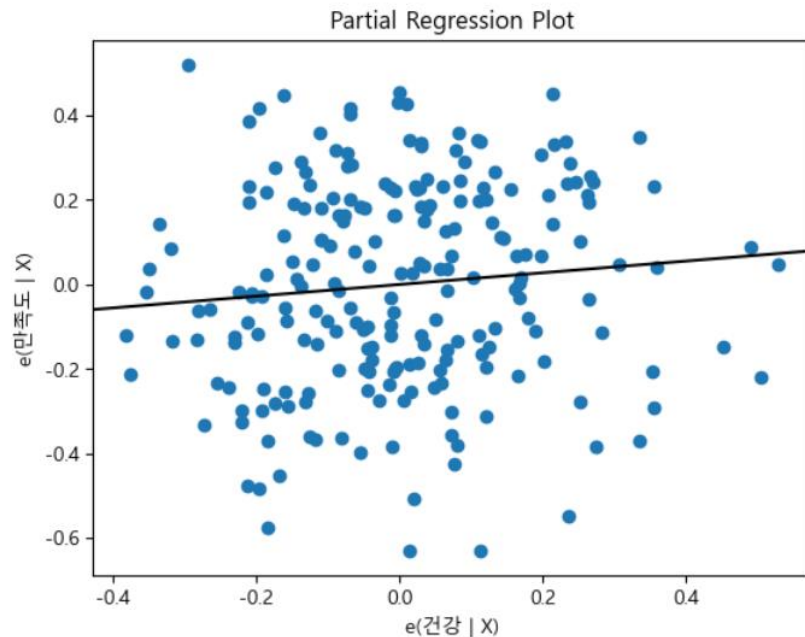
### 3. 통계분석 실습

#### ■ 데이터 분석결과 시각화

- 차트에 한글 적용 및 회귀분석 시각화 라이브러리 import

```
import statsmodels.api as sm  
plt.rcParams['font.family']="Malgun Gothic"  
plt.rcParams['axes.unicode_minus']=False
```

- 만족도(삶의 만족도)와 건강 필드 사이의 연관관계 회귀분석결과 시각화 차트를 작성하라.

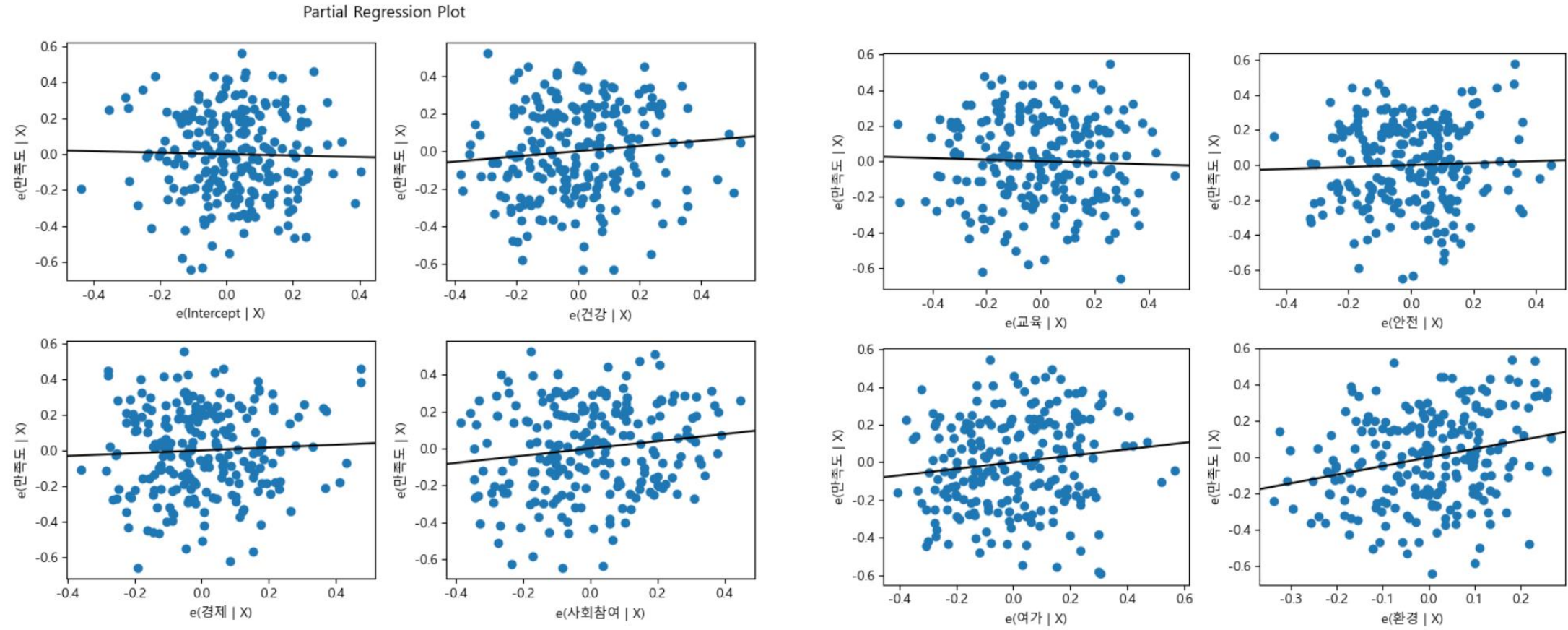




### 3. 통계분석 실습

#### ■ 데이터 분석결과 시각화

- 만족도(삶의 만족도)와 회귀분석에 참여한 모든 필드 사이의 연관관계 회귀분석결과 시각화 차트를 작성하라.

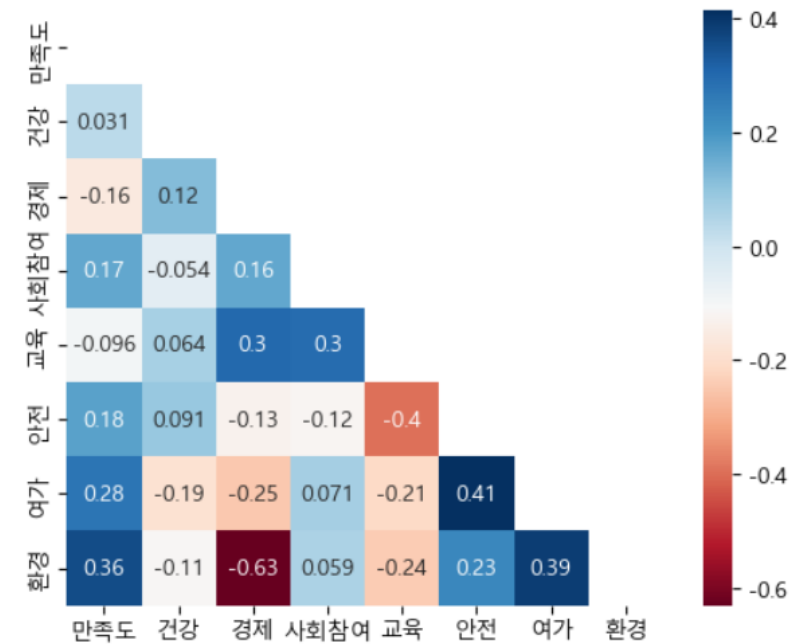


### 3. 통계분석 실습

#### ■ 데이터 분석(상관분석)

- 회귀분석에 사용한 data로 각 필드 사이의 상관계수를 구하고 상관계수 테이블을 사용하여 heatmap를 그림과 같이 작성하라.

	만족도	건강	경제	사회참여	교육	안전	여가	환경
만족도	1.000000	0.030889	-0.159564	0.165182	-0.095661	0.175283	0.275589	0.357748
건강	0.030889	1.000000	0.118797	-0.054158	0.063818	0.090740	-0.186737	-0.112172
경제	-0.159564	0.118797	1.000000	0.163335	0.302246	-0.129147	-0.247433	-0.630315
사회참여	0.165182	-0.054158	0.163335	1.000000	0.295712	-0.118827	0.070937	0.059245
교육	-0.095661	0.063818	0.302246	0.295712	1.000000	-0.396170	-0.213909	-0.241198
안전	0.175283	0.090740	-0.129147	-0.118827	-0.396170	1.000000	0.414465	0.233188
여가	0.275589	-0.186737	-0.247433	0.070937	-0.213909	0.414465	1.000000	0.386590
환경	0.357748	-0.112172	-0.630315	0.059245	-0.241198	0.233188	0.386590	1.000000



Q&A