8과 [예제] 숫자 vs 숫자

1.환경준비

- 라이브러리 불러오기

In [1]:

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

- 데이터 불러오기 : 다음의 예제 데이터를 사용합니다.

> ① 다이아몬드 데이터
> ② 보스톤 시, 타운별 집값
> ③ 아이리스 꽃 분류
> ④ 뉴욕 공기 오염도

In [68]:

```
# 다이아몬드
diamond = pd.read_csv('https://raw.githubusercontent.com/DA4BAM/dataset/master/diamonds
diamond = diamond.sample(3000, random_state = 2022)
diamond.head()
```

Out[68]:

	carat	cut	color	clarity	depth	table	price	x	y	z
50989	0.31	Ideal	G	VS2	61.6	55.0	544	4.37	4.39	2.70
42221	0.33	Ideal	E	IF	62.1	55.0	1289	4.43	4.46	2.76
42307	0.41	Ideal	F	VVS1	62.1	57.0	1295	4.75	4.79	2.96
27207	2.02	Very Good	F	SI1	62.7	59.0	17530	7.97	8.03	5.02
22207	1.50	Good	H	VS1	63.4	59.0	10256	7.20	7.29	4.59

In [3]:

```
# 아이리스 꽃 분류
iris = pd.read_csv('https://raw.githubusercontent.com/DA4BAM/dataset/master/iris.csv')
iris.head()
```

Out[3]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [4]:

```
# 보스톤 집값 데이터
boston = pd.read_csv('https://raw.githubusercontent.com/DA4BAM/dataset/master/boston2_N
boston.head()
```

Out[4]:

	crim	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv	zn2
0	0.00632	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0	1.(
1	0.02731	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6	0.(
2	0.02729	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7	0.(
3	0.03237	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4	0.(
4	0.06905	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2	0.(

In [5]:

```
# 뉴욕시 공기 오염도 데이터
air = pd.read_csv('https://raw.githubusercontent.com/DA4BAM/dataset/master/air2.csv')
air['Date'] = pd.to_datetime(air['Date'])
air['Month'] = air.Date.dt.month
air['Weekday'] = air.Date.dt.weekday
air.head()
```

Out[5]:

	Ozone	Solar.R	Wind	Temp	Date	Month	Weekday
0	41	190.0	7.4	67	1973-05-01	5	1
1	36	118.0	8.0	72	1973-05-02	5	2
2	12	149.0	12.6	74	1973-05-03	5	3
3	18	313.0	11.5	62	1973-05-04	5	4
4	19	NaN	14.3	56	1973-05-05	5	5

2.시각화

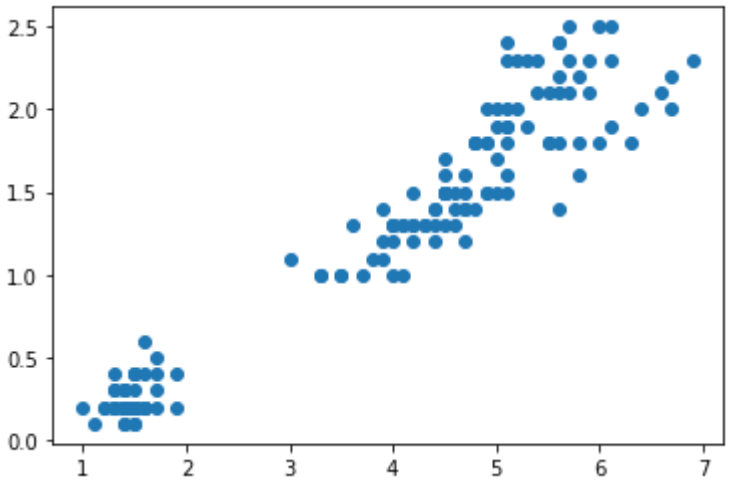
1) 산점도 : plt.scatter, sns.scatterplot, sns.jointplot

① iris의 Petal.Length와 Petal.Width의 관계를 살펴보기 위해 산점도를 그려봅시다.

- plt.scatter

In [28]:

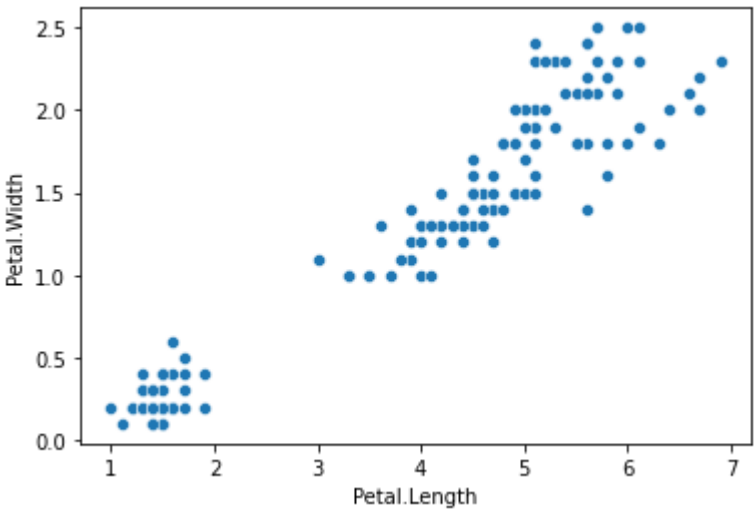
```
1 plt.scatter(iris['Petal.Length'], iris['Petal.Width'])
2 plt.show()
```



- sns.scatterplot

In [31]:

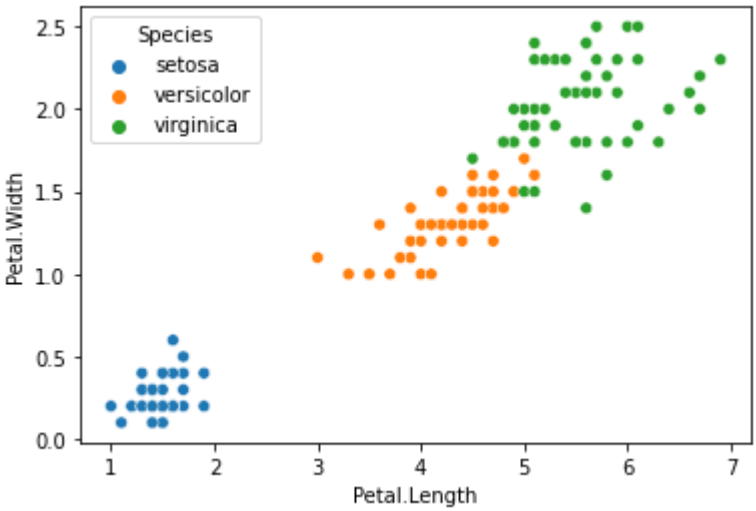
```
1 sns.scatterplot(x='Petal.Length', y='Petal.Width', data = iris)
2 plt.show()
```



- sns.scatterplot(, hue = 'Species')

In [32]:

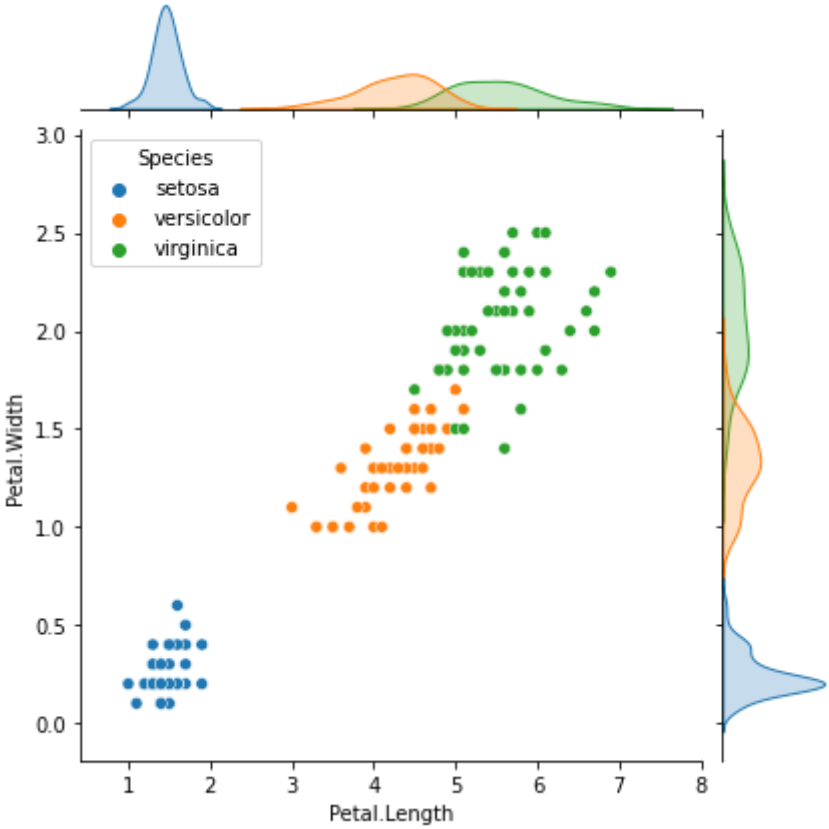
```
sns.scatterplot(x='Petal.Length', y='Petal.Width', data = iris, hue = 'Species')
plt.show()
```



- sns.joinplot

In [38]:

```
sns.jointplot(x='Petal.Length', y='Petal.Width', data = iris, hue = 'Species')
plt.show()
```



- 그래프로 부터 파악된 내용을 적어 봅시다.

In [ ]:

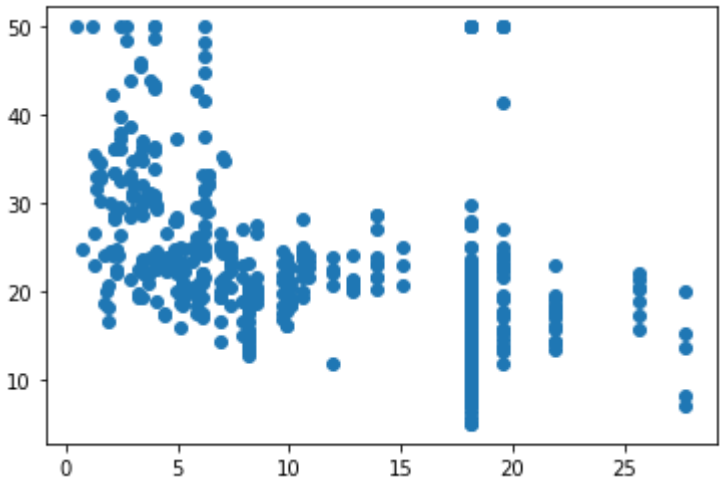
```
1
```

② boston의 indus(비소매상업지구의 면적비율)와 medv(집값)의 관계를 살펴보기 위해 산점도를 그려봅시다.

- plt.scatter

In [35]:

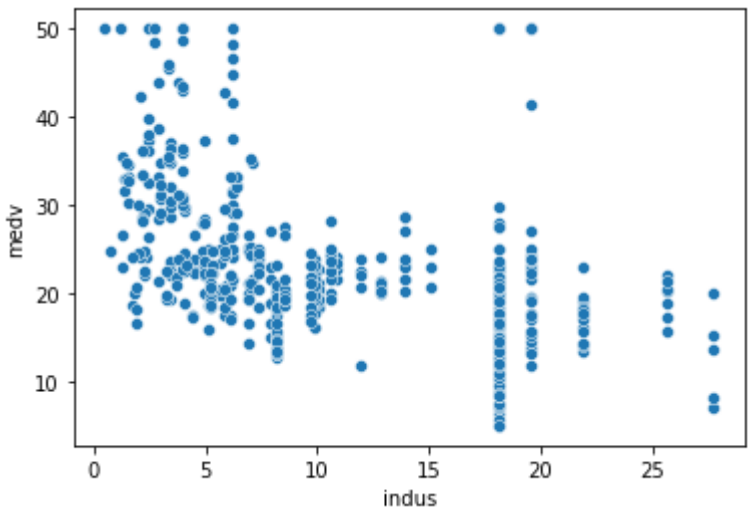
```
1 plt.scatter(boston['indus'], boston['medv'])
2 plt.show()
```



- sns.scatterplot

In [36]:

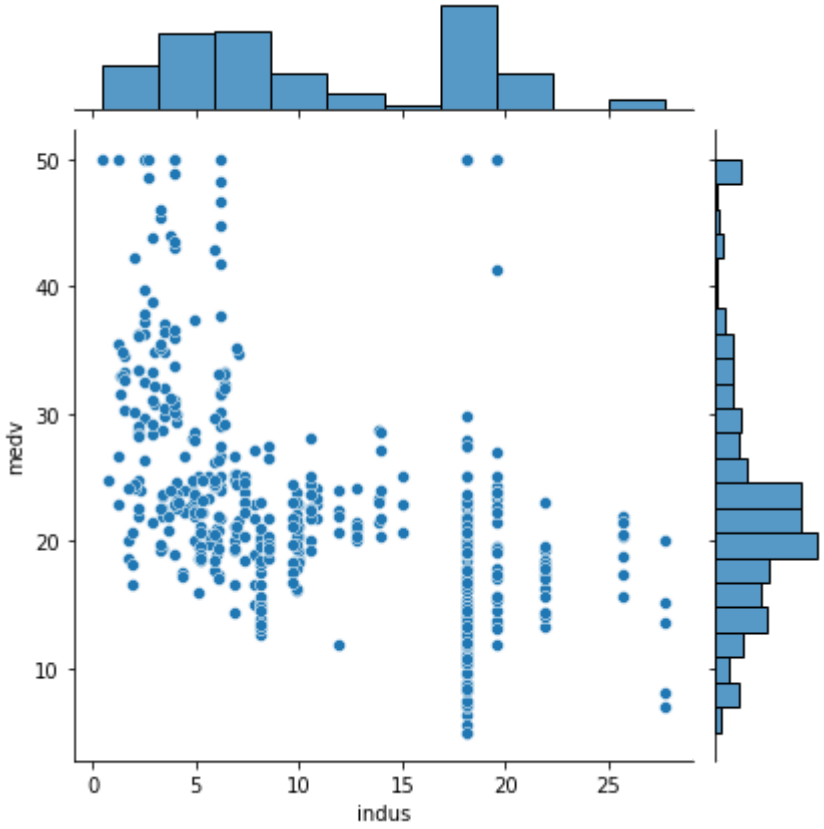
```
1 sns.scatterplot(x='indus', y='medv', data = boston)
2 plt.show()
```



- sns.jointplot

In [39]:

```
1 sns.jointplot(x='indus', y='medv', data = boston)
2 plt.show()
```



- 그래프로 부터 파악된 내용을 적어 봅시다.

In [ ]:

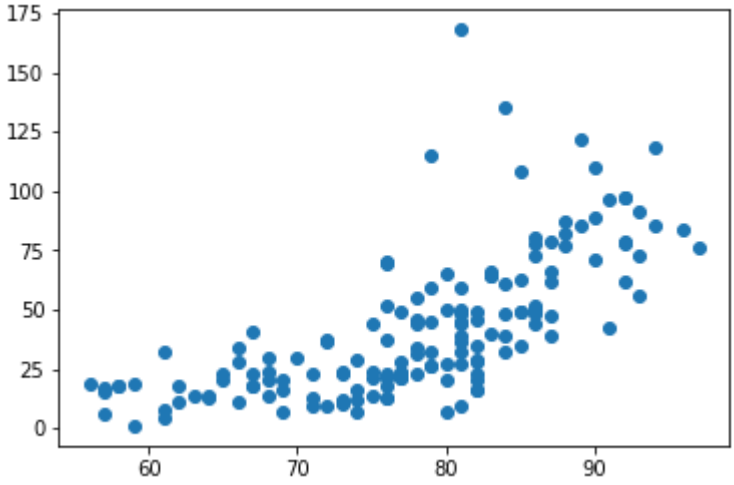
```
1
```

③ air의 Temp와 Ozone의 관계를 살펴보기 위해 산점도를 그려봅시다.

- plt.scatter

In [40]:

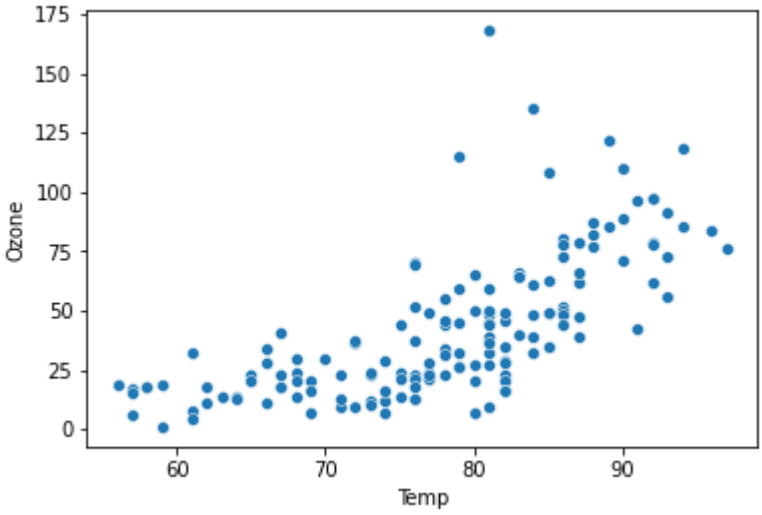
```
plt.scatter(air['Temp'], air['Ozone'])
plt.show()
```



- sns.scatterplot

In [41]:

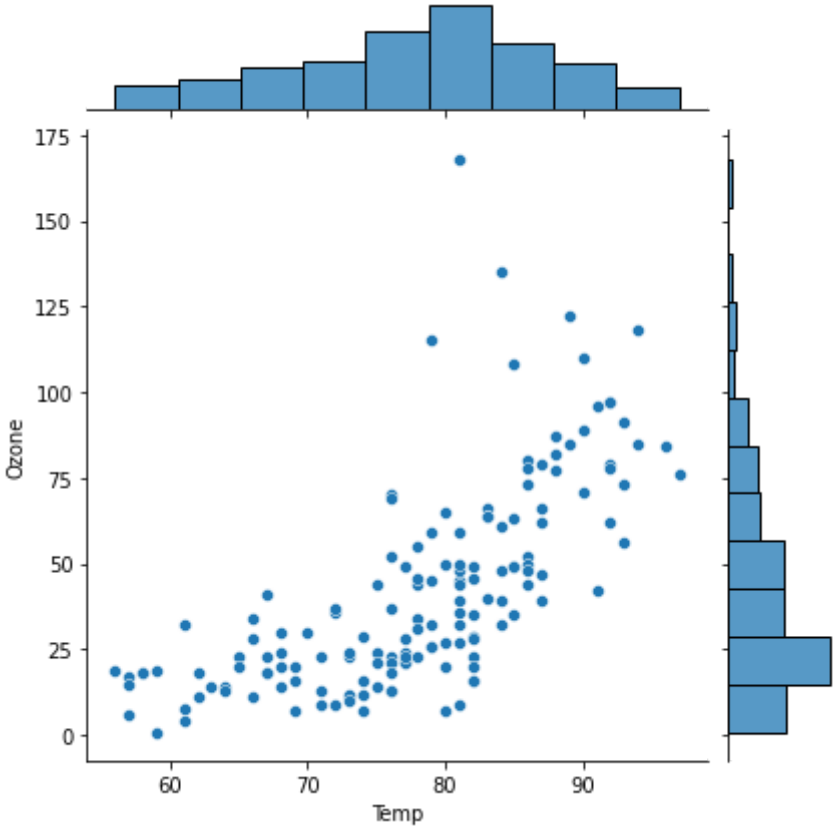
```
sns.scatterplot(x='Temp', y='Ozone', data = air)
plt.show()
```



- sns.jointplot

In [42]:

```
1 sns.jointplot(x='Temp', y='Ozone', data = air)
2 plt.show()
```



- 그래프로 부터 파악된 내용을 적어 봅시다.

In [ ]:

```
1
```

In [ ]:

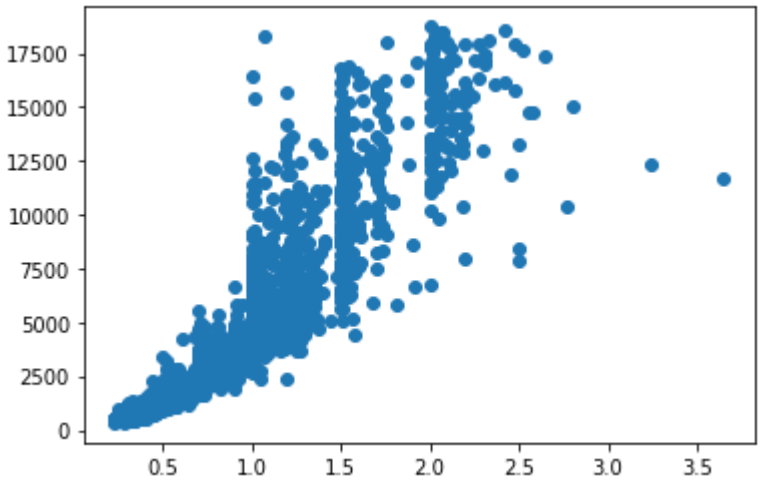
```
1
```

④ diamond의 carat과 price의 관계를 살펴보기 위해 산점도를 그려봅시다.

- plt.scatter

In [65]:

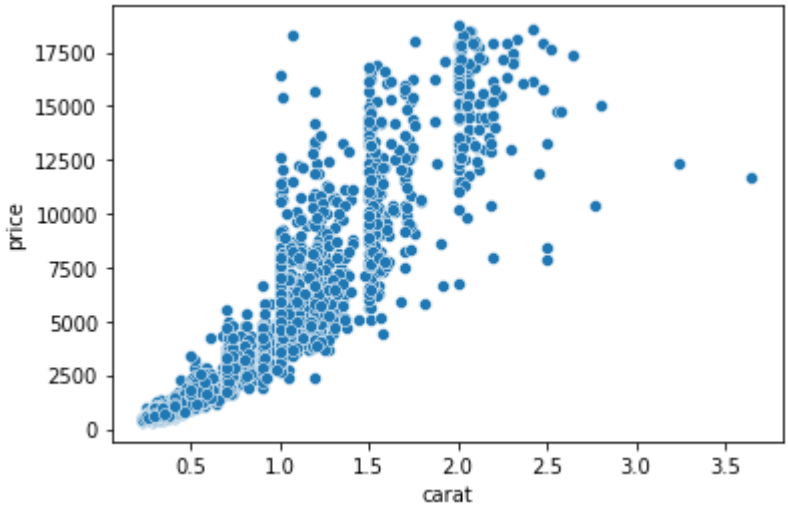
```
plt.scatter(diamond['carat'], diamond['price'])
plt.show()
```



- sns.scatterplot

In [66]:

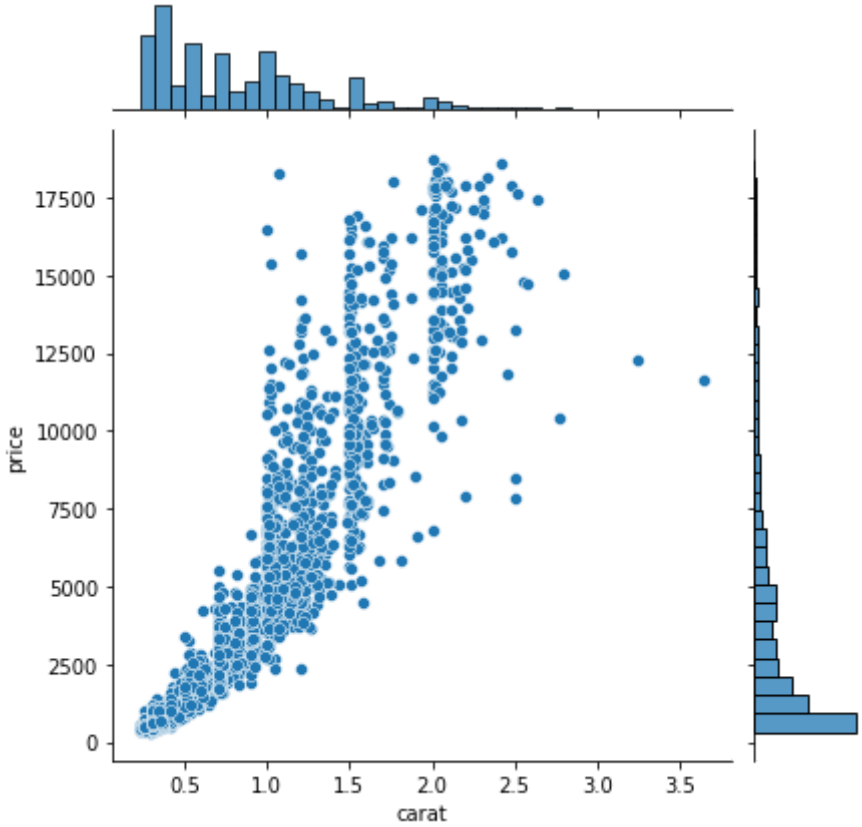
```
sns.scatterplot(x='carat', y='price', data = diamond)
plt.show()
```



- sns.jointplot

In [67]:

```
sns.jointplot(x='carat', y='price', data = diamond)
plt.show()
```



- 그래프로 부터 파악된 내용을 적어 봅시다.

In [ ]:

```
```

In [ ]:

```
```

2) 한꺼번에 산점도 그리기 : sns.pairplot

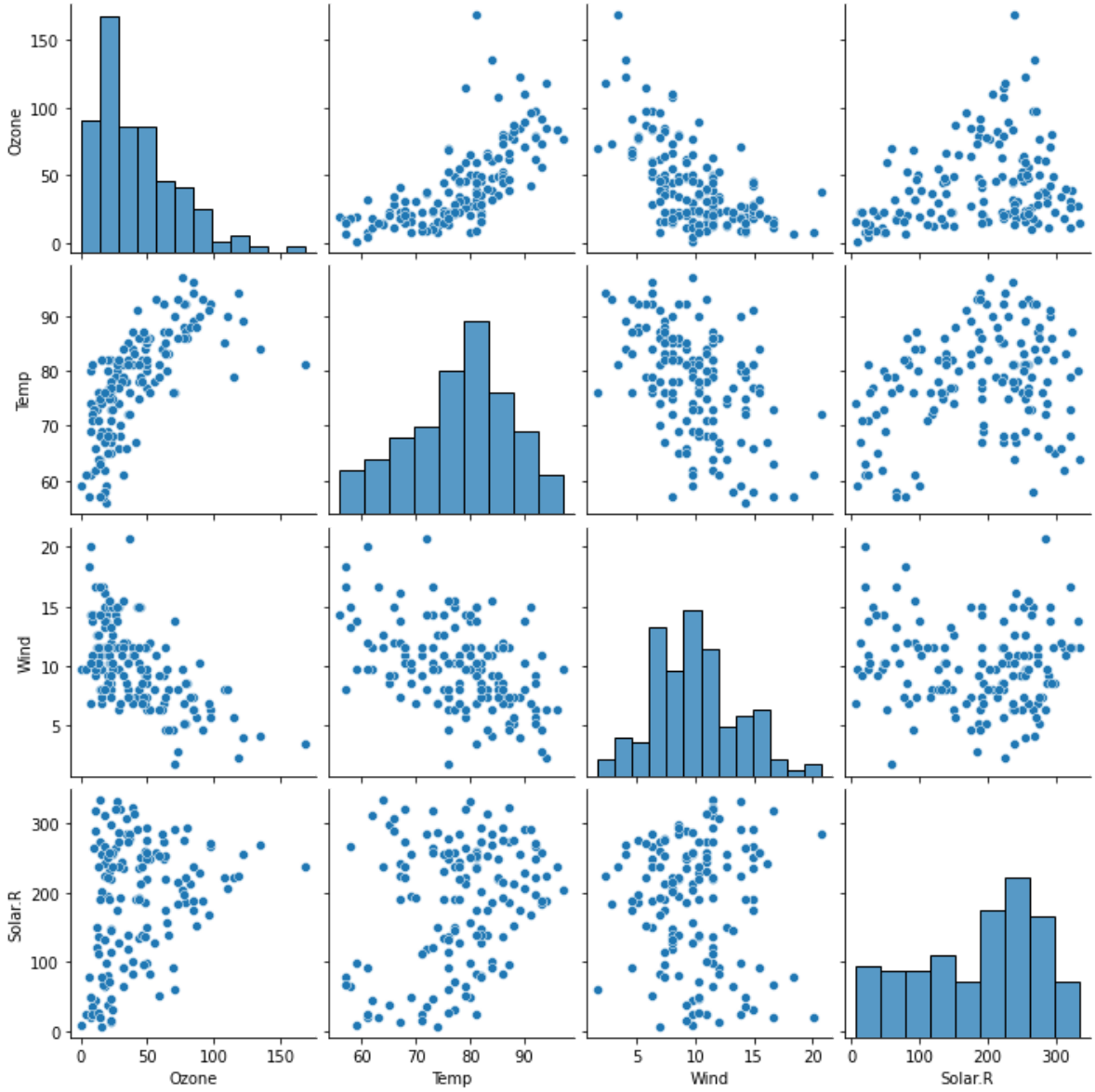
① air 데이터프레임에서 Month와 Day를 제외하고 산점도를 한꺼번에 그려봅시다.

In [45]:

```
1 sns.pairplot(air.loc[:, ['Ozone','Temp','Wind','Solar.R']])
2 plt.show()
```

Out[45]:

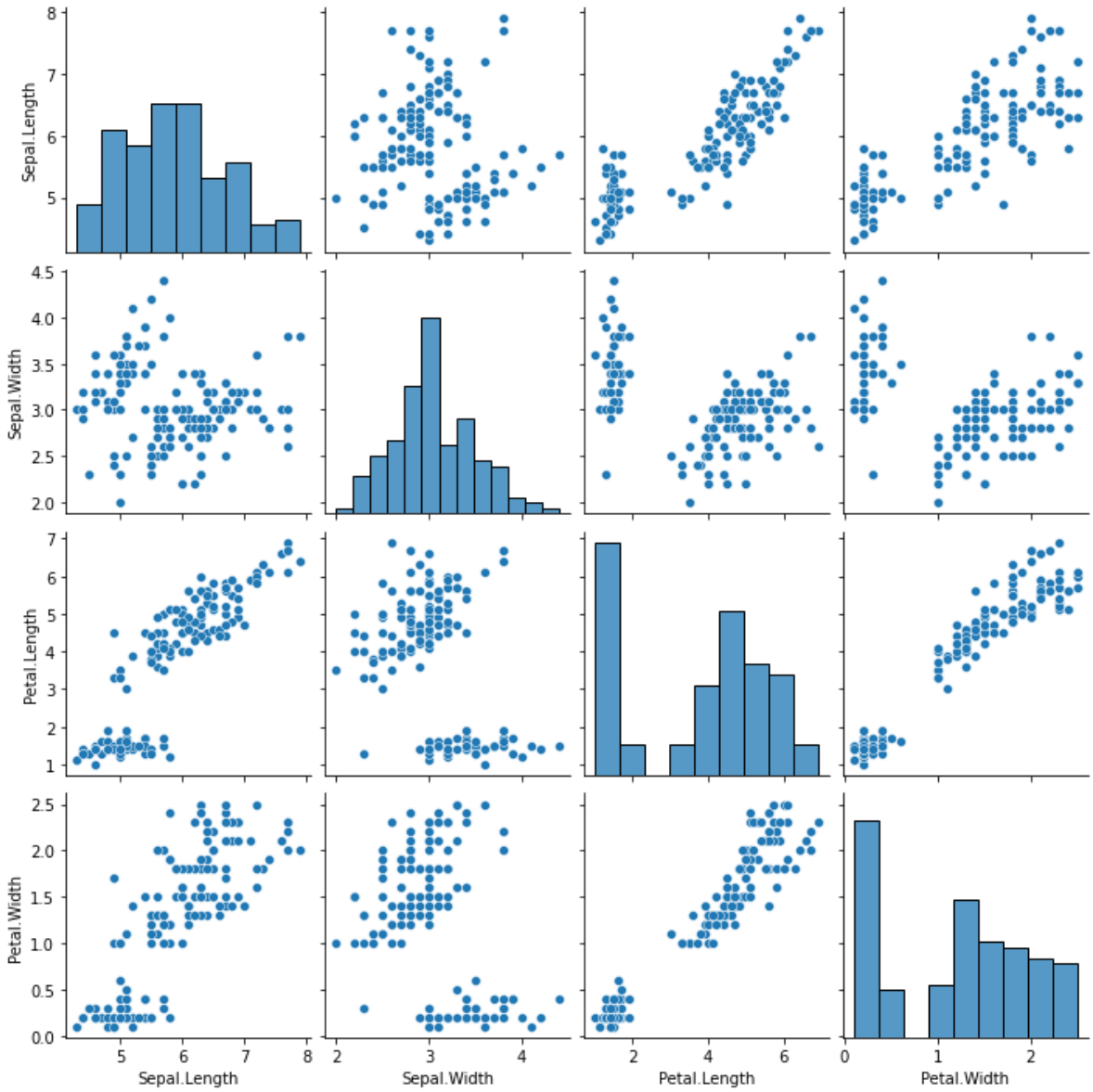
```
<seaborn.axisgrid.PairGrid at 0x7f8b01b32290>
```



② iris 데이터프레임에 대해서, Species를 제외하고 한꺼번에 산점도를 그려봅시다.

In [50]:

```
1 sns.pairplot(iris.loc[:, ['Sepal.Length','Sepal.Width', 'Petal.Length', 'Petal.Width']]
2 plt.show()
```



3.수치화 : 상관분석

In [52]:

```
1 import scipy.stats as spst
```

① boston.indus와 boston.medv의 관계를 수치화 해 봅시다.

결과를 해석해 봅시다.

In [54]:

```
1 # 상관계수와 p-value
2 spst.pearsonr(boston['indus'], boston['medv'])
```

Out[54]:

```
(-0.483725160028373, 4.900259981751351e-31)
```

In [ ]:

```
1
```

② diamond의 각 변수들 간에 상관 계수를 구해 봅시다.

In [69]:

```
1 diamond.corr()
```

Out[69]:

	carat	depth	table	price	x	y	z
carat	1.000000	0.027081	0.160081	0.918257	0.978485	0.976614	0.962422
depth	0.027081	1.000000	-0.305373	-0.016220	-0.027121	-0.030945	0.091622
table	0.160081	-0.305373	1.000000	0.097818	0.172048	0.166498	0.130414
price	0.918257	-0.016220	0.097818	1.000000	0.884365	0.886169	0.866249
x	0.978485	-0.027121	0.172048	0.884365	1.000000	0.997668	0.980012
y	0.976614	-0.030945	0.166498	0.886169	0.997668	1.000000	0.980099
z	0.962422	0.091622	0.130414	0.866249	0.980012	0.980099	1.000000

③ 위 결과에서 가장 강한 상관관계와 약한 상관관계를 찾아 봅시다.

In [ ]:

```
1
```