

Pandas 시각화



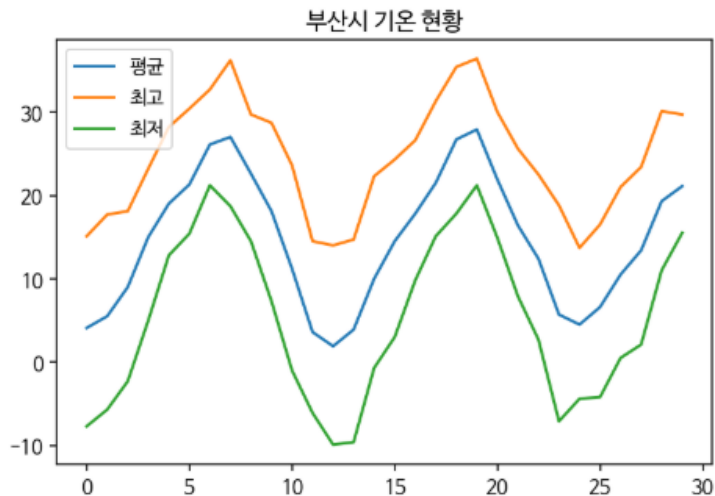
matplotlib 라이브러리 사용

- 꺾은 선그래프
 - `df.plot()`
- 막대 그래프
 - `df.plot(kind='bar')`
- 히스토 그래프
 - `df.plot.hist()`
- 산점도 그래프
 - `df.plot.scatter()`

찍은 선그래프

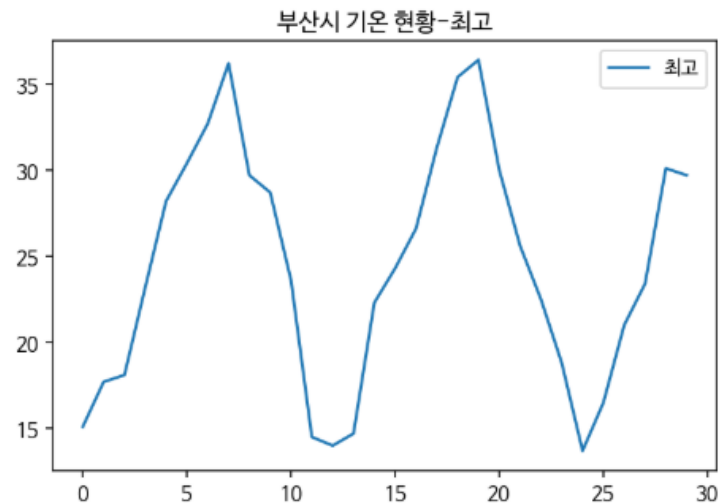
#기본 그래프

```
df.plot()  
plt.title('부산시 기온 현황')  
plt.legend()  
plt.show()
```



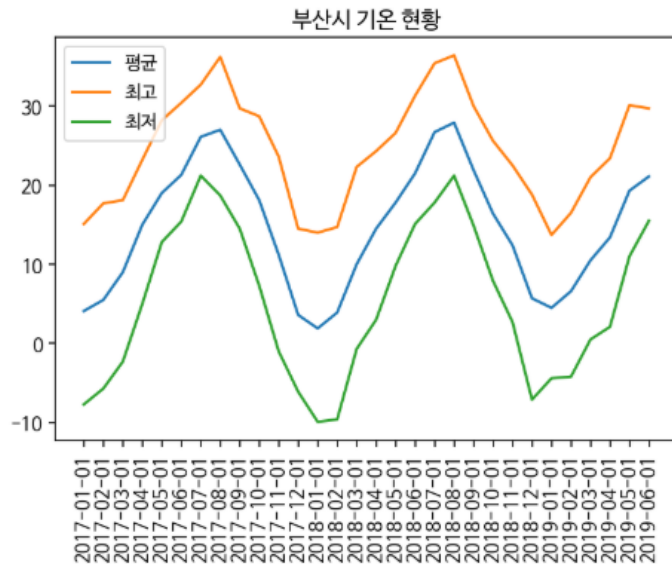
#특정 항목 그래프

```
df['최고'].plot()  
plt.title('부산시 기온 현황-최고')  
plt.legend()  
plt.show()
```



찍은 선그래프

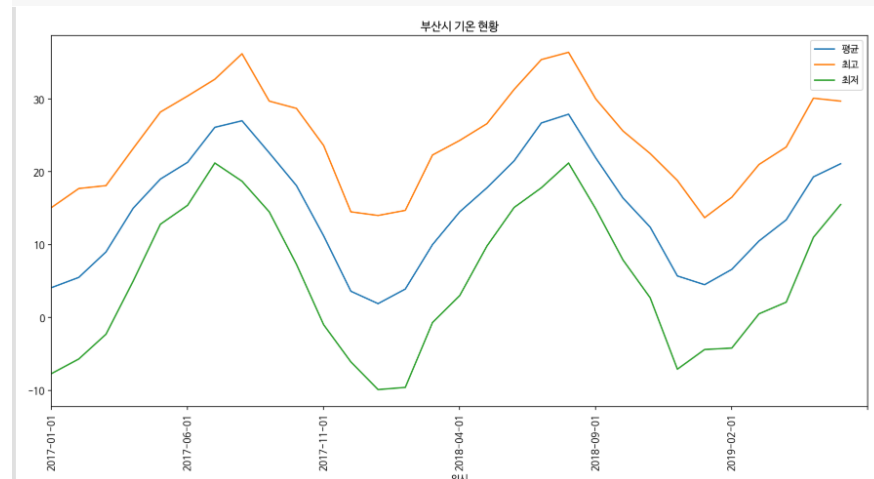
```
#x축지정하기
df.plot()
plt.title('부산시 기온 현황')
plt.legend()
plt.xticks(df.index, df['일시'], rotation=90)
plt.show()
```



```
#인덱스를 변경하여 작성
df = df.set_index('일시')
```

```
#그림 크기 변경
df.plot(figsize=(15, 7))

plt.title('부산시 기온 현황')
plt.legend()
plt.xlim(0, len(df.index))
plt.xticks(rotation=90)
plt.show()
```



subplot

```
fig = plt.figure(figsize=(15,10))

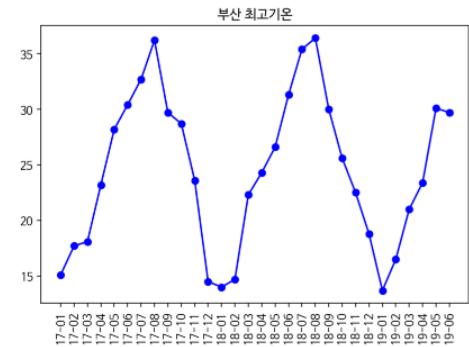
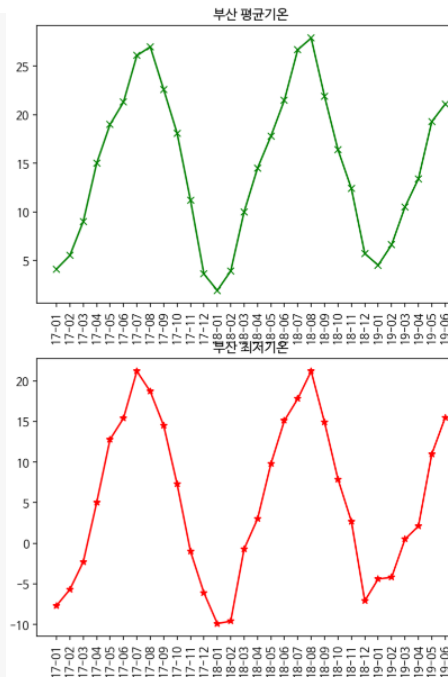
ax1 = fig.add_subplot(2,2,1)
ax2 = fig.add_subplot(2,2,2)
ax3 = fig.add_subplot(2,2,3)

ax1.plot(df['평균'], 'gx-')
ax1.set_title('부산 평균기온')
ax1.set_xticklabels(df.index.str[2:7], rotation=90)

ax2.plot(df['최고'], 'bo-')
ax2.set_title('부산 최고기온')
ax2.set_xticklabels(df.index.str[2:7], rotation=90)

ax3.plot(df['최저'], 'r*-')
ax3.set_title('부산 최저기온')
ax3.set_xticklabels(df.index.str[2:7], rotation=90)

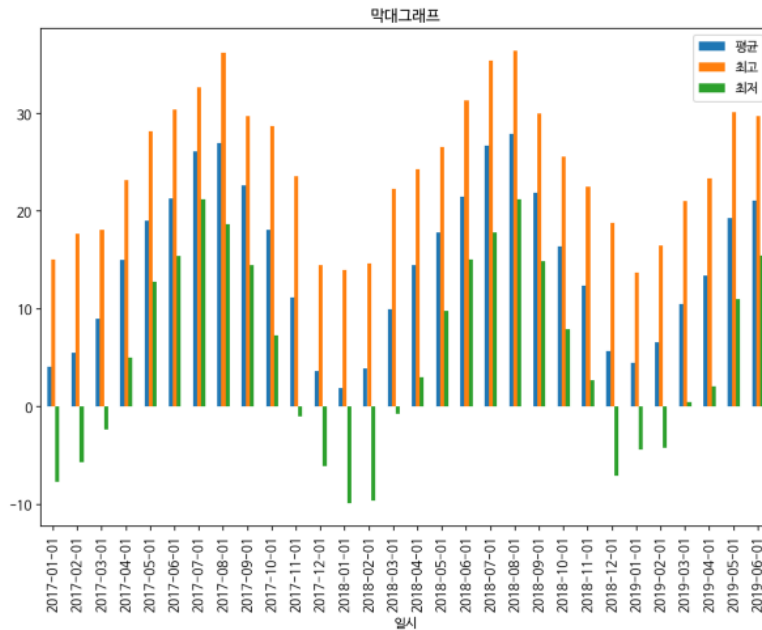
plt.show()
```



막대 그래프

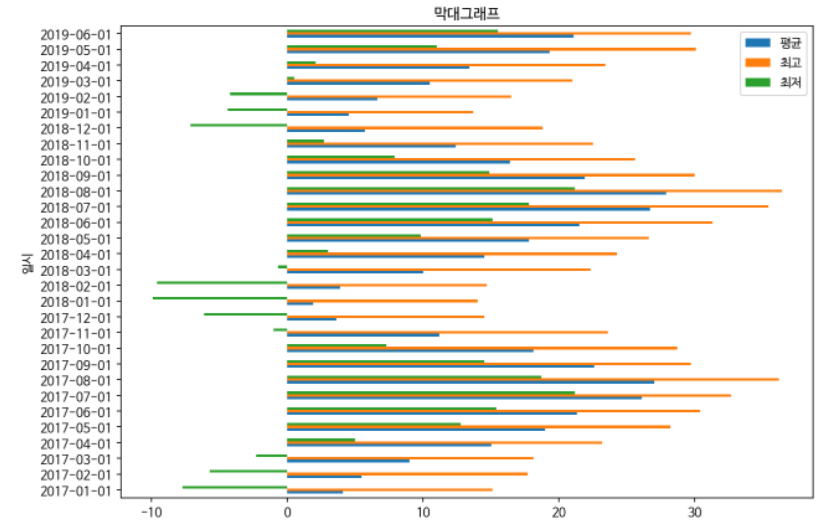
#세로 막대형

```
df.plot(kind='bar', figsize=(10,7))
plt.title('막대그래프')
plt.show()
```



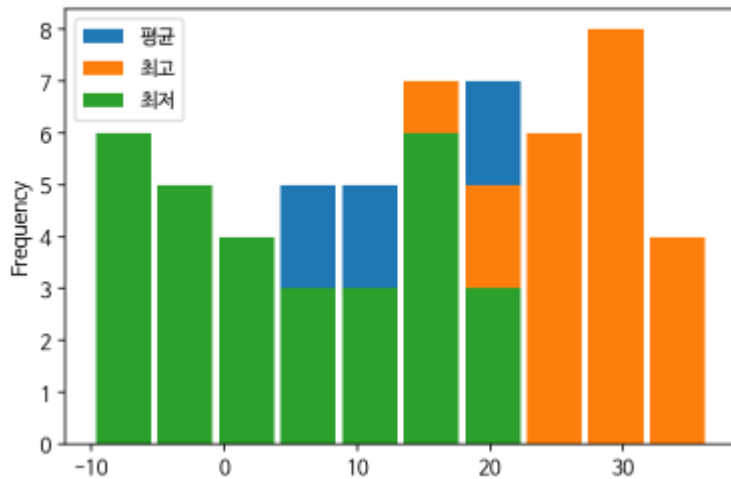
#가로 막대형

```
df.plot(kind='barh', figsize=(10,7))
plt.title('막대그래프')
plt.show()
```



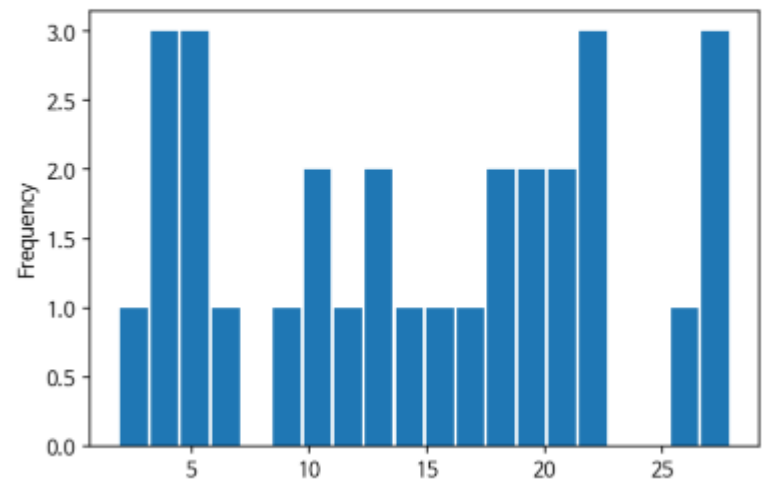
히스토그램

```
df.plot.hist(rwidth=0.9)  
plt.show()
```



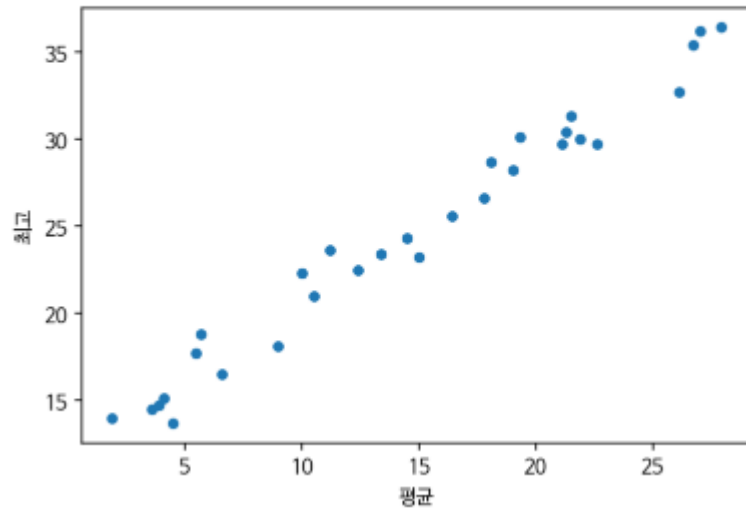
#히스토그램 그리기

```
df['평균'].plot.hist(rwidth=0.9, bins=20)  
plt.show()
```

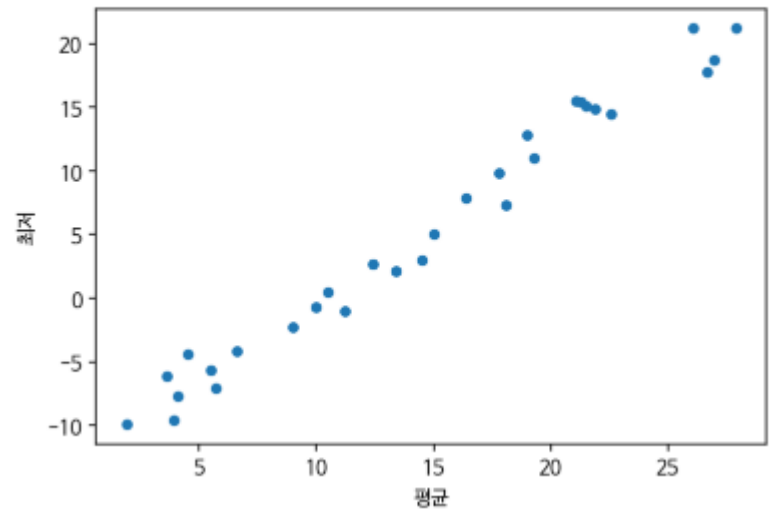


산점도 그래프

```
df.plot.scatter(x='평균', y='최고')  
plt.show()
```

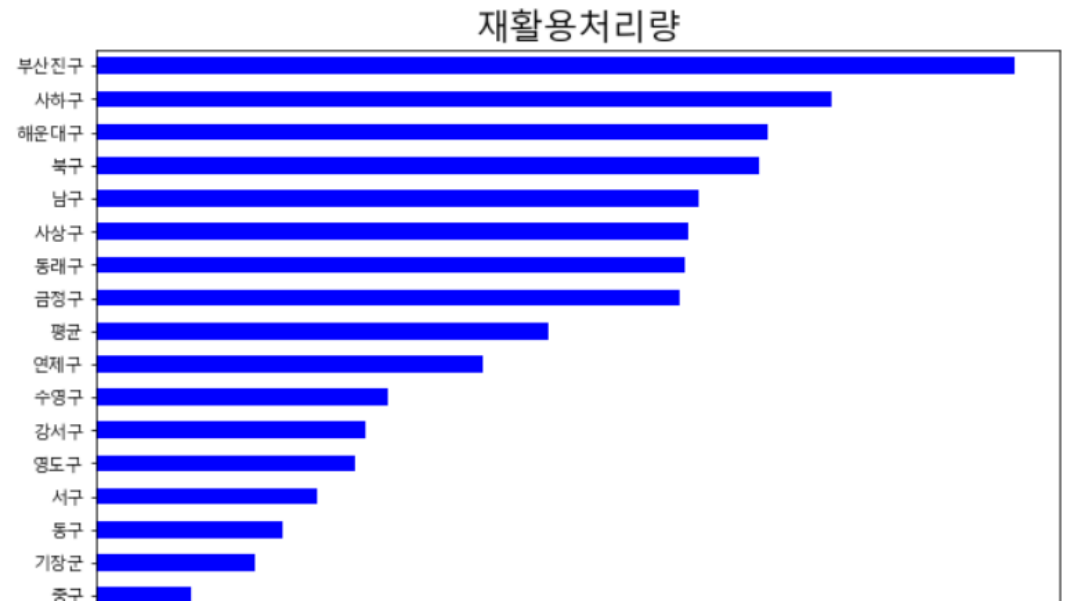


```
df.plot.scatter(x='평균', y='최저')  
plt.show()
```



해결문제

- 2017년 부산쓰레기발생 자료를 이용하여 재활용 처리량이 많은 지역 순서대로 수평 막대 그래프를 그리시오.



seaborn 라이브러리 사용

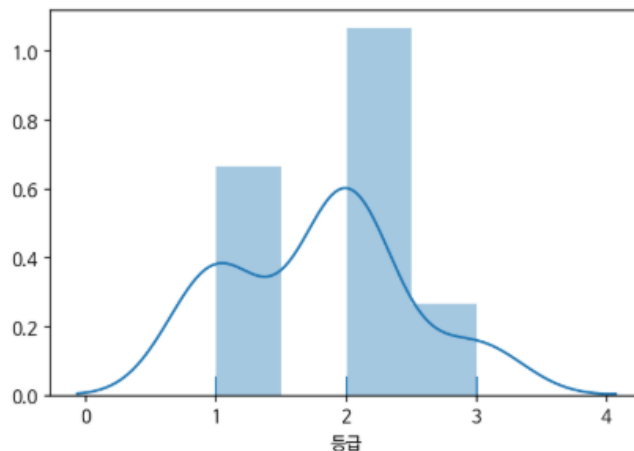
- **import seaborn as sns**
 - Matplotlib을 기반으로 다양한 색상 테마와 통계용 차트 등의 기능을 추가한 시각화 패키지
 - 기본적인 시각화 기능은 Matplotlib 패키지에 따르며 통계 기능은 Statsmodels 패키지를 사용
 - <http://seaborn.pydata.org/>



.distplot

- `sns.distplot(x, kde=True, rug=True)`
 - 1차원 실수 분포 플롯
 - 러그와 커널 밀도 표시 기능이 있어서 Matplotlib의 hist 명령보다 많이 사용
 - 러그 : 데이터 위치를 x축 위에 작은 선분(rug)으로 나타내어 실제 데이터들의 위치를 보여준
 - 커널 밀도 : 커널이라는 함수를 겹치는 방법으로 히스토그램보다 부드러운 형태의 분포 곡선을 보여주는 방법

```
sns.distplot(df['등급'], kde=True, rug=True, hist=True)  
plt.show()
```

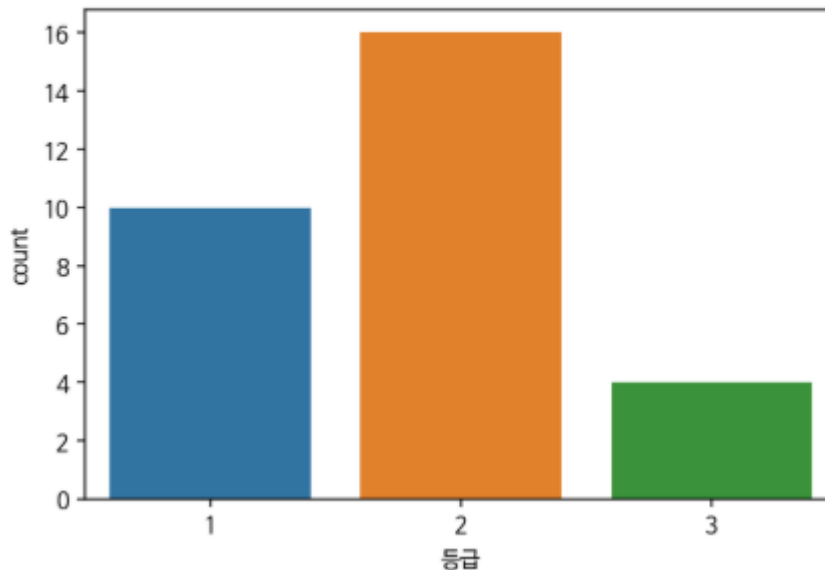


.countplot()

- **sns.countplot(x)**

- 각 카테고리 값별로 데이터가 얼마나 있는지 표시

```
sns.countplot(df['등급'])  
plt.show()
```

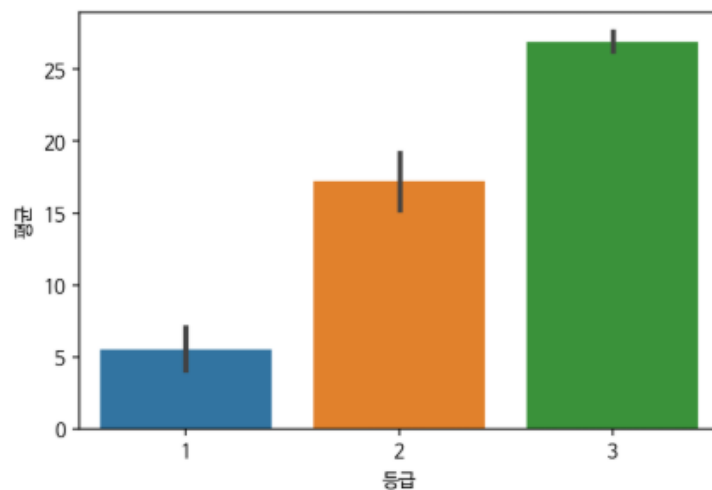


.barplot()

- `sns.barplot(x=열, y=열, data=데이터프레임)`

– 각 카테고리 값별로 y에 지정한 평균을 계산하여 보여줌

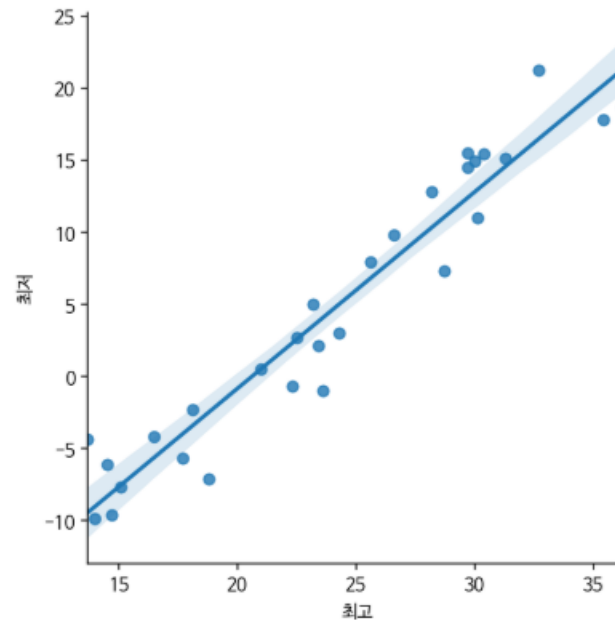
```
sns.barplot(x='등급', y='평균', data=df)  
plt.show()
```



.Implot()

- `sns.Implot(x=열1, y=열2, data=데이터 프레임)`
 - 선형관계확인

```
sns.Implot(x='최고', y='최저', data=df)  
plt.show()
```

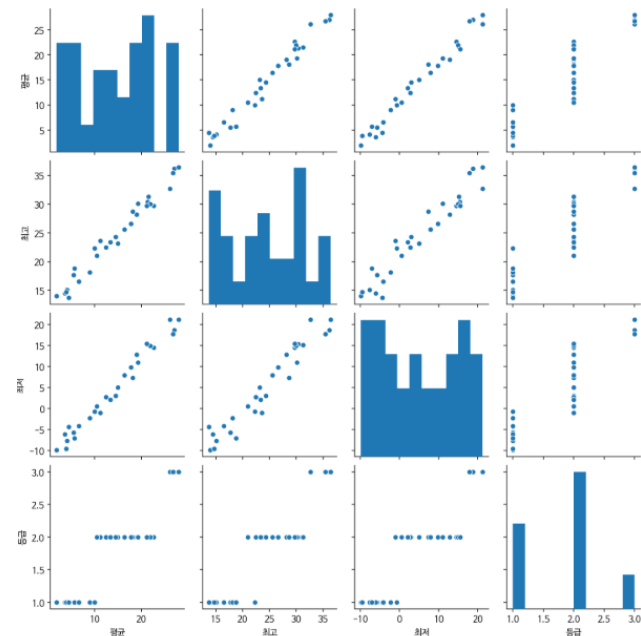


.pairplot()

- **sns.pairplot(데이터프레임)**

- 각 데이터 열의 조합에 대해 스캐터 플롯
- 같은 데이터가 만나는 대각선 영역에는 해당 데이터의 히스토그램

```
sns.pairplot(df)  
plt.show()
```

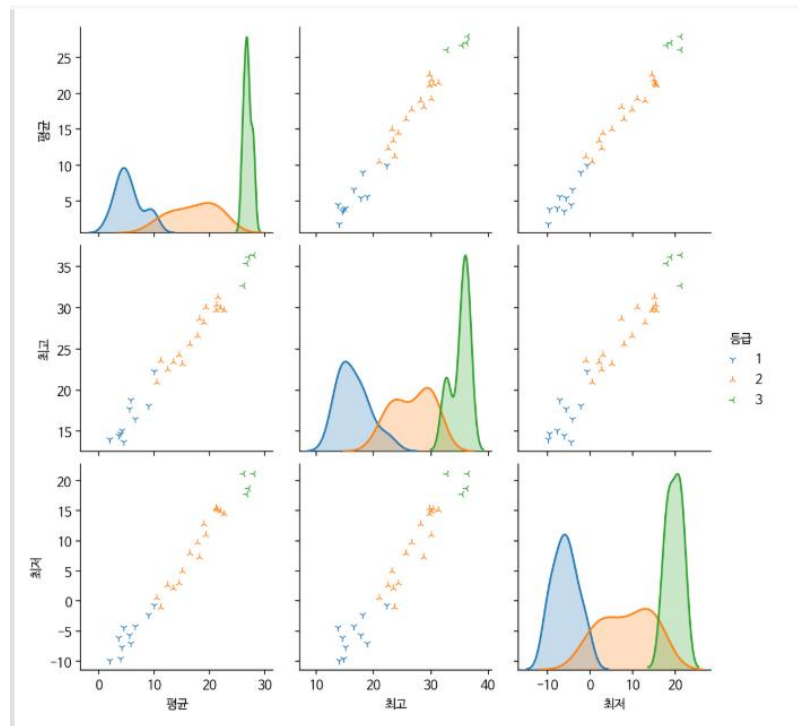


.pairplot()

- **sns.pairplot(데이터프레임)**

- 카테고리형 데이터가 섞여 있는 경우에는 hue 인수에 카테고리 변수 이름을 지정하여 카테고리 값에 따라 색상을 다르게 할 수 있음

```
sns.pairplot(df, hue="등급", markers=["1", "2", "3"])  
plt.show()
```



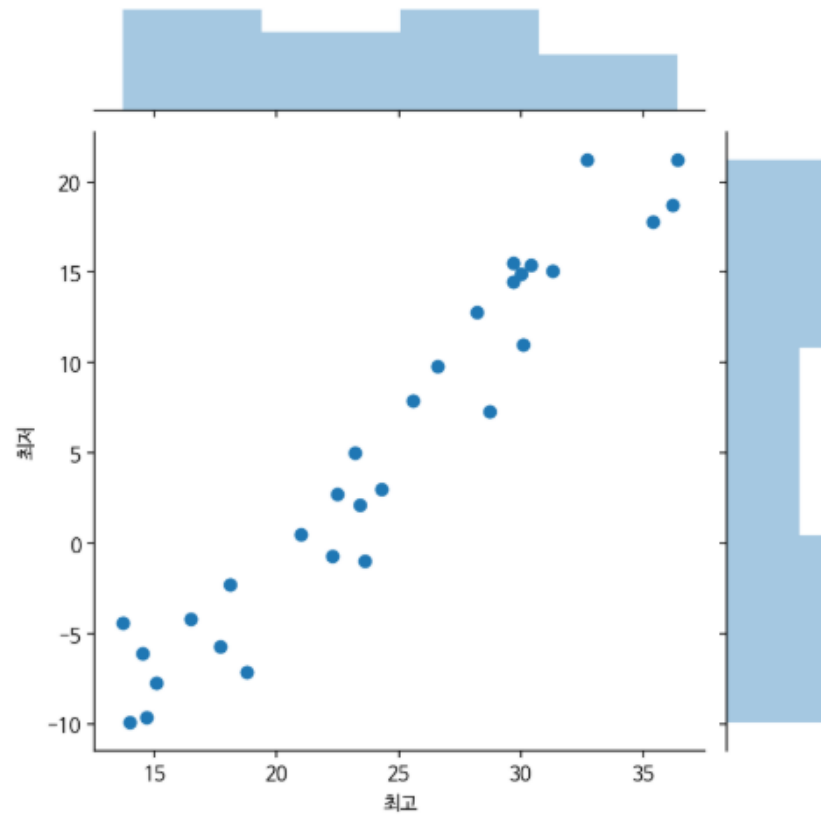
.jointplot()

- `sns.jointplot(x="x_name", y="y_name", data=dataframe, kind='scatter')`
 - 데이터가 2차원이고 모두 연속적인 실수값이라면 스캐터 플롯(scatter plot)을 사용
 - 같은 데이터가 만나는 대각선 영역에는 해당 데이터의 히스토그램
 - 스캐터 플롯뿐 아니라 차트의 가장자리(margin)에 각 변수의 히스토그램
 - `kind='scatter'` : 스캐터 플롯
 - `kind='kde'` : 커널 밀도 히스토그램
 - `kind='hex'` : 육각그래프



.jointplot()

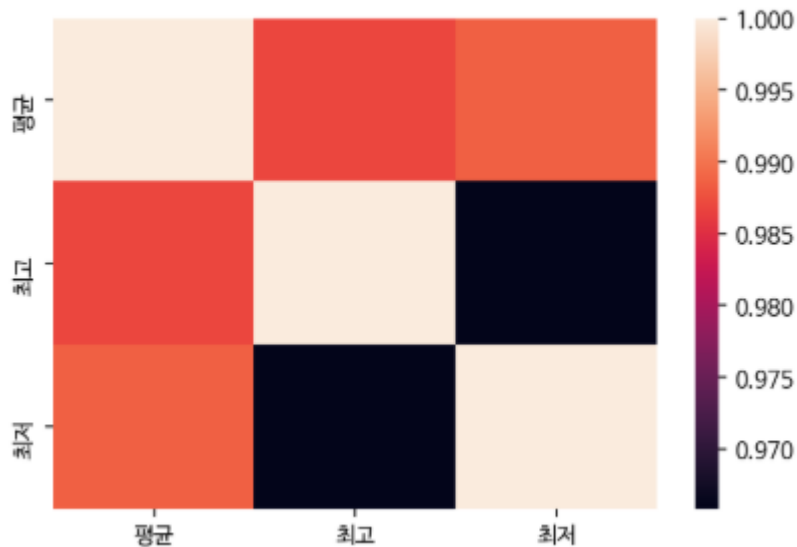
```
sns.jointplot(x="최고", y="최저", data=df, kind='scatter')  
plt.show()
```



.heatmap()

- 상관관계 분석

```
sns.heatmap(df.corr())  
plt.show()
```



해결문제

- 부산재활용처리량.xlsx 자료를 이용하여 부산시의 재활용처리량 그래프를 작성하시오.

