Ch12 Interactive Graph

마우스의 움직임에 실시간으로 반응하는 그래프 만들기

> 그래프를 자유롭게 제어하면서 관심있는 부분을 자세히 살펴볼 수 있음

[사전 준비] 관련 패키지 설치

(Anaconda Prompt에서)

> pip install plotly
>
> pip install jupyter-dash

패키지 설치 후 Jupyter Lab 재실행

In [2]:

```
import pandas as pd
df_exam = pd.read_csv("Test_result_CSV.csv", encoding = 'UTF-8')
df_exam
```

Out[2]:

	Unnamed: 0	ID	Name	Class	Attendance	Homework	Midterm	Final
0	0	2022001	고ㅇㅇ	B	14	17	22	22
1	1	2022002	석ㅇㅇ	A	18	20	27	25
2	2	2022003	강ㅇㅇ	A	20	19	22	25
3	3	2022004	민ㅇㅇ	A	19	19	24	26
4	4	2022005	지ㅇㅇ	A	20	19	24	23
...	...	...	...	...	...	...	...	...
69	69	2022070	유ㅇㅇ	B	18	19	20	23
70	70	2022071	김ㅇㅇ	B	20	19	23	19
71	71	2022072	이ㅇㅇ	B	18	17	23	24
72	72	2022073	은ㅇㅇ	A	14	18	29	26
73	73	2022074	김ㅇㅇ	B	20	17	28	27

74 rows × 8 columns

In [7]:

```
##### 'total' 파생변수를 추가 : 'total'은 총점
df_exam['Total'] = df_exam['Attendance'] + df_exam['Homework'] + df_exam['Midterm']

#### Data Frame에 'grade' 파생변수를 추가
##### > 'Grade'는 총점에 대한 등급 : "A+" : 100-95, "A" : 94-90, "B+" : 89-85, "B" :
#####     "C+" : 79-75, "C" : 74-70, "D" : 69-60, "F" : 59-0
import numpy as np
df_exam['Grade'] = np.where(df_exam['Total'] >= 95, 'A+',
                    np.where(df_exam['Total'] >= 90, 'A',
                    np.where(df_exam['Total'] >= 85, 'B+',
                    np.where(df_exam['Total'] >= 80, 'B',
                    np.where(df_exam['Total'] >= 75, 'C+',
                    np.where(df_exam['Total'] >= 70, 'C',
                           np.where(df_exam['Total'] >= 60, 'D', 'F')))))))
df_exam
```

Out[7]:

	Unnamed: 0	ID	Name	Class	Attendance	Homework	Midterm	Final	Total	Grade
0	0	2022001	고ㅇㅇ	B	14	17	22	22	75	C+
1	1	2022002	석ㅇㅇ	A	18	20	27	25	90	A
2	2	2022003	강ㅇㅇ	A	20	19	22	25	86	B+
3	3	2022004	민ㅇㅇ	A	19	19	24	26	88	B+
4	4	2022005	지ㅇㅇ	A	20	19	24	23	86	B+
...	...	...	...	...	...	...	...	...	...	...
69	69	2022070	유ㅇㅇ	B	18	19	20	23	80	B
70	70	2022071	김ㅇㅇ	B	20	19	23	19	81	B
71	71	2022072	이ㅇㅇ	B	18	17	23	24	82	B
72	72	2022073	은ㅇㅇ	A	14	18	29	26	87	B+
73	73	2022074	김ㅇㅇ	B	20	17	28	27	92	A

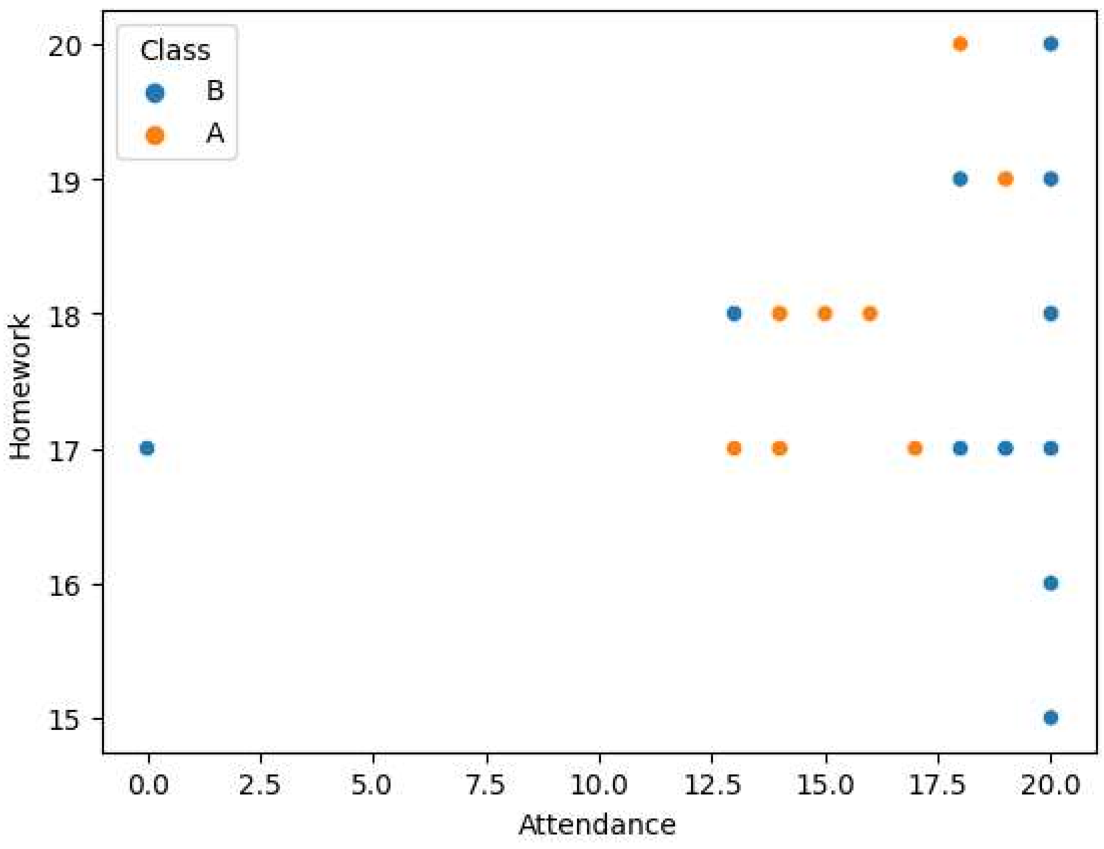
74 rows × 10 columns

In [8]:
```
import seaborn as sns
sns.scatterplot(data = df_exam, x = 'Attendance', y = 'Homework', hue = 'Class')
```

```
C:₩Users₩ADMIN₩anaconda3₩lib₩site-packages₩seaborn₩_oldcore.py:1498: FutureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead

C:₩Users₩ADMIN₩anaconda3₩lib₩site-packages₩seaborn₩_oldcore.py:1498: FutureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead

C:₩Users₩ADMIN₩anaconda3₩lib₩site-packages₩seaborn₩_oldcore.py:1498: FutureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead

C:₩Users₩ADMIN₩anaconda3₩lib₩site-packages₩seaborn₩_oldcore.py:1498: FutureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead

C:₩Users₩ADMIN₩anaconda3₩lib₩site-packages₩seaborn₩_oldcore.py:1498: FutureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
```

Out[8]:

```
<Axes: xlabel='Attendance', ylabel='Homework'>
```



In [9]:

```python
## 반응형 산점도 그래프
import plotly.express as px
px.scatter(data_frame = df_exam, x = 'Attendance', y = 'Homework', color = 'Class')
```

In [13]:
```
## 그룹핑 데이터프레임 구성
df_grp = df_exam.groupby('Grade', as_index = False)  ₩
                .agg(cnt_grade = ('Grade', 'count'))

## 그룹핑 결과에 비율(ratio) 열 추가
df_grp['ratio'] = df_grp['cnt_grade'] * 100 / sum(df_grp['cnt_grade'])
df_grp
```

Out[13]:

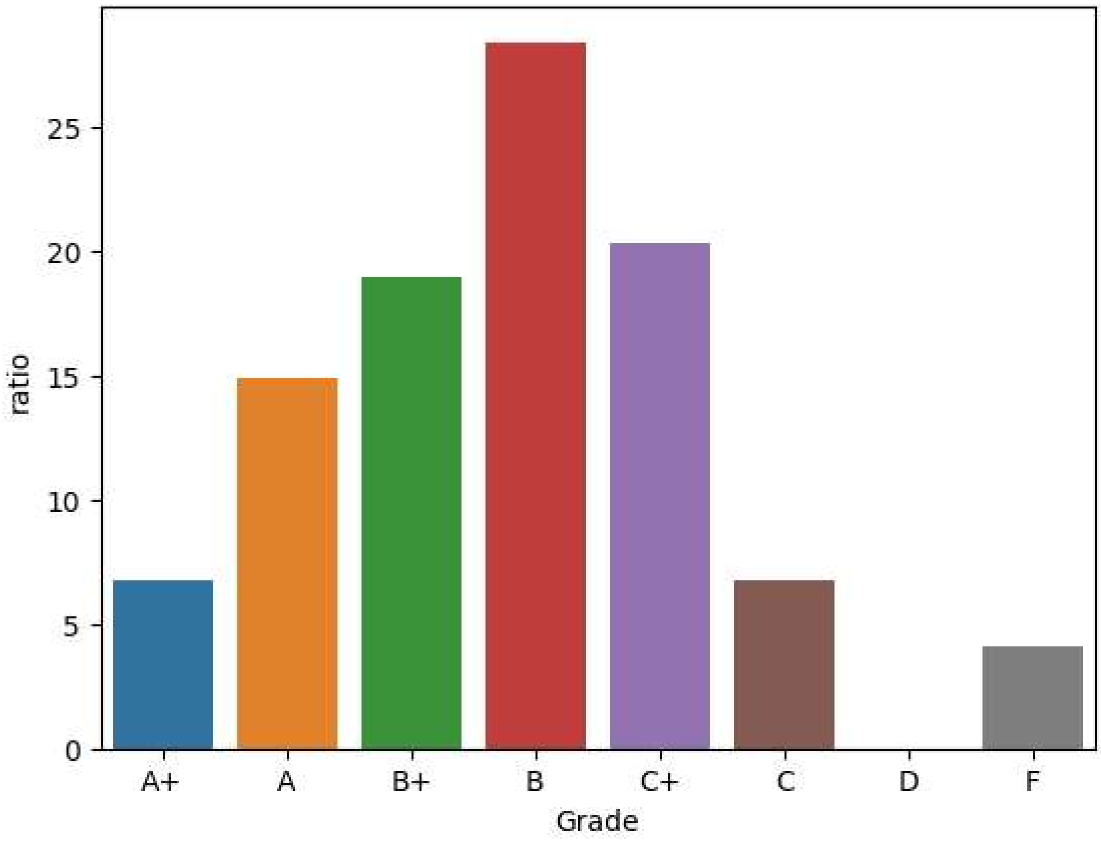
	Grade	cnt_grade	ratio
0	A	11	14.864865
1	A+	5	6.756757
2	B	21	28.378378
3	B+	14	18.918919
4	C	5	6.756757
5	C+	15	20.270270
6	F	3	4.054054

In [15]:
```
## 막대 그래프 그리기
grade_order = ['A+', 'A', 'B+', 'B', 'C+', 'C', 'D', 'F']
sns.barplot(data = df_grp, x = 'Grade', y = 'ratio', order = grade_order)
```

```
C:₩Users₩ADMIN₩anaconda3₩lib₩site-packages₩seaborn₩_oldcore.py:1498: FutureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead

C:₩Users₩ADMIN₩anaconda3₩lib₩site-packages₩seaborn₩_oldcore.py:1498: FutureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isin
stance(dtype, CategoricalDtype) instead
```

Out[15]:

```
<Axes: xlabel='Grade', ylabel='ratio'>
```



In [17]:

```
## 반응형 막대 그래프 그리기
px.bar(data_frame = df_grp, x = 'Grade', y = 'ratio', color = 'Grade')
```

```
In [18]: ## 등급별 비율 데이터프레임 생성
         df_grp = df_exam.groupby('Grade', as_index = False)  ₩
                         .agg(cnt_grade = ('Grade', 'count'))
         df_grp['ratio'] = df_grp['cnt_grade'] * 100 / sum(df_grp['cnt_grade'])
         df_grp
```

```
Out[18]:    Grade  cnt_grade      ratio
         0      A         11  14.864865
         1     A+          5   6.756757
         2      B         21  28.378378
         3     B+         14  18.918919
         4      C          5   6.756757
         5     C+         15  20.270270
         6      F          3   4.054054
```

```
In [19]: ## 대화형 선 그래프 그리기
         px.line(df_grp, x = 'Grade', y = 'ratio')
```

```
In [20]: ## 대화형 박스 그래프 그리기
         px.box(df_exam, x = 'Class', y = 'Total', color = 'Class', width = 600, height = 40(
```

그래프를 HTML로 저장하기

.write_html()

.open_new()

In [21]:
```
## 표를 HTML 파일로 저장하기
fig = px.box(df_exam, x = 'Class', y = 'Total', color = 'Class', width = 600, height
fig.write_html('box_plot.html')
```

In [22]:
```
# html 문서 열기
import webbrowser as wb
wb.open_new('box_plot.html')
```

Out[22]:
```
True
```

In [ ]: