[4-3. Seaborn 소개 및 사용 방법]

1. Seaborn 소개

Seaborn은 Python에서 사용할 수 있는 데이터 시각화 라이브러리로, Matplotlib을 기반으로 하여 더 고급 그래픽과 통계적 그래프를 쉽게 그릴 수 있도록 설계되었습니다. Seaborn은 Michael Waskom에 의해 개발되었으며, 2012년경 첫 출시되었습니다.

Seaborn은 Matplotlib 위에 구축되어 Matplotlib의 기능을 모두 사용할 수 있지만, 기본 설정과 디자인이 개선되어 보다 현대적이고 아름다운 시각화를 쉽게 만들 수 있습니다. Seaborn은 복잡한 데이터의 패턴을 시각화하고, 여러 변수간의 관계를 분석하며, 데이터의 분포를 탐색하는 데 특히 유용합니다.

Seaborn을 사용하기 위해서는 라이브러리가 설치되어 있어야 합니다. 아나콘다 네비게이터 "Environments" 탭에서 현재 사용 중인 가상환경을 선택한 후 라이브러리 이름을 조회하여 설치할 수 있습니다.

커맨드 라인(명령 프롬프트, 터미널 등) 환경에서는 "pip install seaborn" 명령어를 사용해서 설치할 수 있습니다.

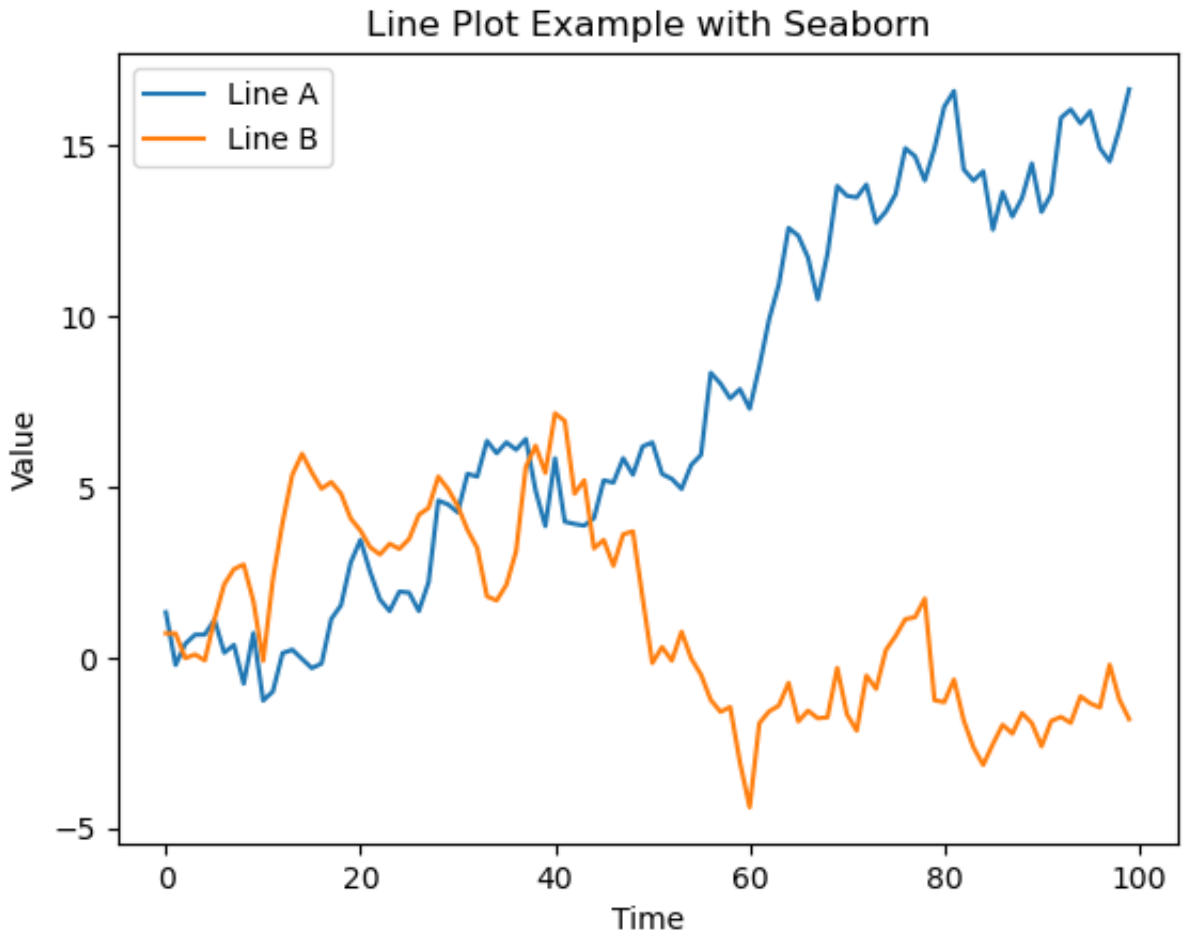
2. 기본적인 선그래프 생성

Seaborn에서는 lineplot 함수를 사용하여 시간에 따른 데이터의 트렌드나 연속적인 데이터의 변화를 시각화할 수 있습니다.

Seaborn을 사용하여 그래프를 그리는 경우에도, 보다 세밀한 그래프의 조정이나 추가적인 설정을 위해서는 matplotlib.pyplot 모듈을 함께 사용하는 것이 일반적입니다. 이러한 설정에는 그래프의 제목 설정, 축 레이블 설정, 범례 위치 조정, 축 범위 설정 등이 포함될 수 있습니다.

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# 예시 데이터 생성
np.random.seed(10)
data = pd.DataFrame(
    data=np.random.randn(100, 2),
    columns=["A", "B"]
).cumsum()
data['Time'] = pd.Series(list(range(len(data))))

# 선 그래프 그리기
sns.lineplot(x='Time', y='A', data=data, label='Line A')
sns.lineplot(x='Time', y='B', data=data, label='Line B')

# 그래프 제목 및 레이블 설정
plt.title('Line Plot Example with Seaborn')
plt.xlabel('Time')
plt.ylabel('Value')

# 범례 표시
plt.legend()

plt.show()
```

```
/Users/KP_Hong/anaconda3/envs/fastcampus/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/KP_Hong/anaconda3/envs/fastcampus/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/KP_Hong/anaconda3/envs/fastcampus/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/KP_Hong/anaconda3/envs/fastcampus/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



[경고문구 내용 확인]

You

나는 코드에서 mode.use_inf_as_na를 사용하지 않았는데, 어디에서 문제가 생긴거고, 어떻게 해결하면 되지?

‹ 2 / 2 ›

Professional Coder (Auto programming)

`mode.use_inf_as_na` 경고는 직접적으로 당신의 코드에서 발생한 것이 아닙니다. 이 경고는 Seaborn 라이브러리 내부에서 발생하는 것으로, Seaborn이 내부적으로 Pandas 옵션 `mode.use_inf_as_na`를 사용하고 있기 때문입니다. Pandas에서 이 옵션은 더 이상 사용되지 않을 예정이며, 그로 인해 해당 경고가 발생합니다.

문제 원인

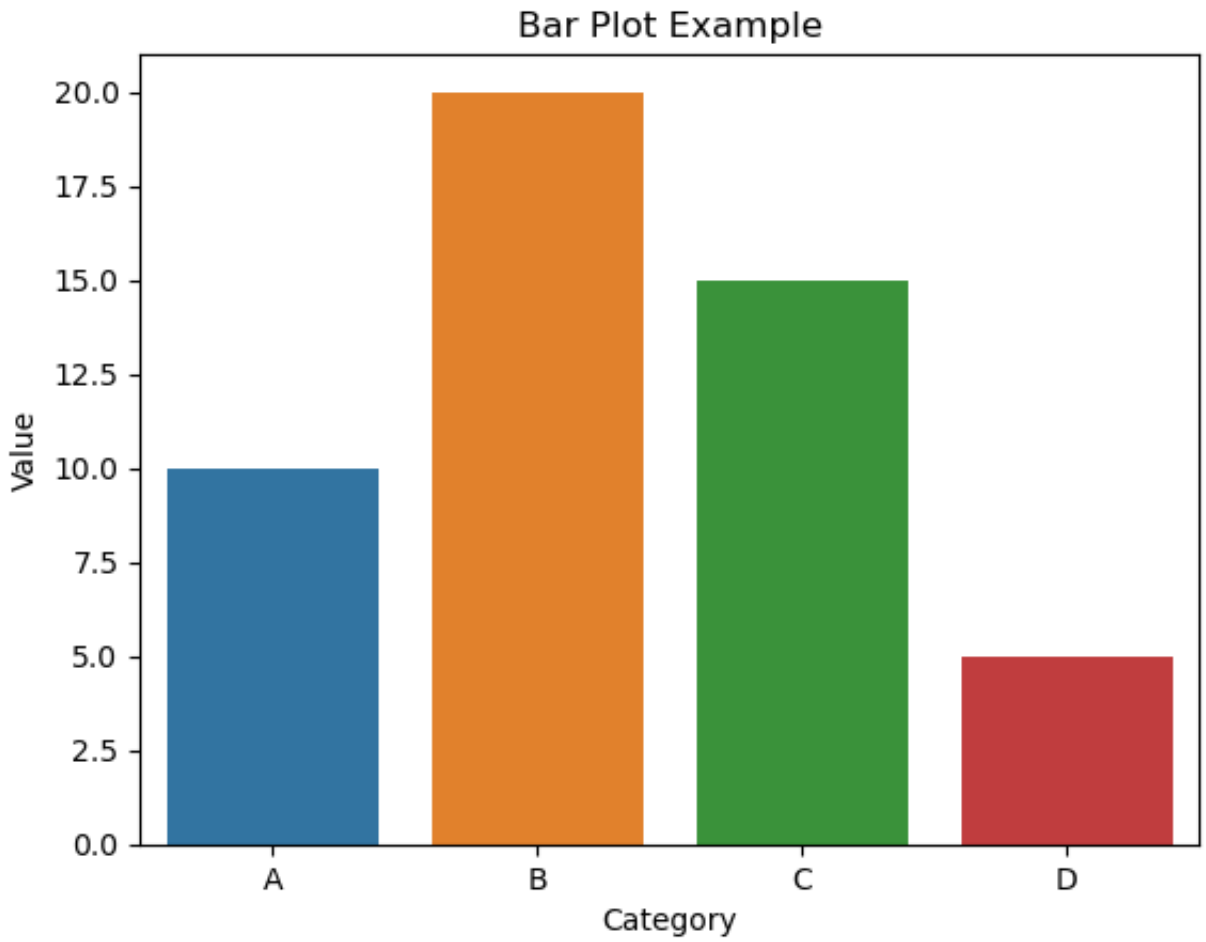
문제의 원인은 Seaborn이 여전히 Pandas의 곧 제거될 예정인 기능을 사용하고 있기 때문입니다. 이는 Seaborn 라이브러리가 아직 최신 Pandas API 변경 사항에 완전히 대응하지 못했음을 의미합니다.

3. 막대그래프 그리기

3-1. barplot 사용하기

barplot은 x축의 범주별로 y축의 수치 데이터(또는 데이터의 평균(기본 설정))을 막대로 표현합니다.
또한, 신뢰 구간을 error bar로 표시하여 데이터의 분산을 나타낼 수 있습니다.

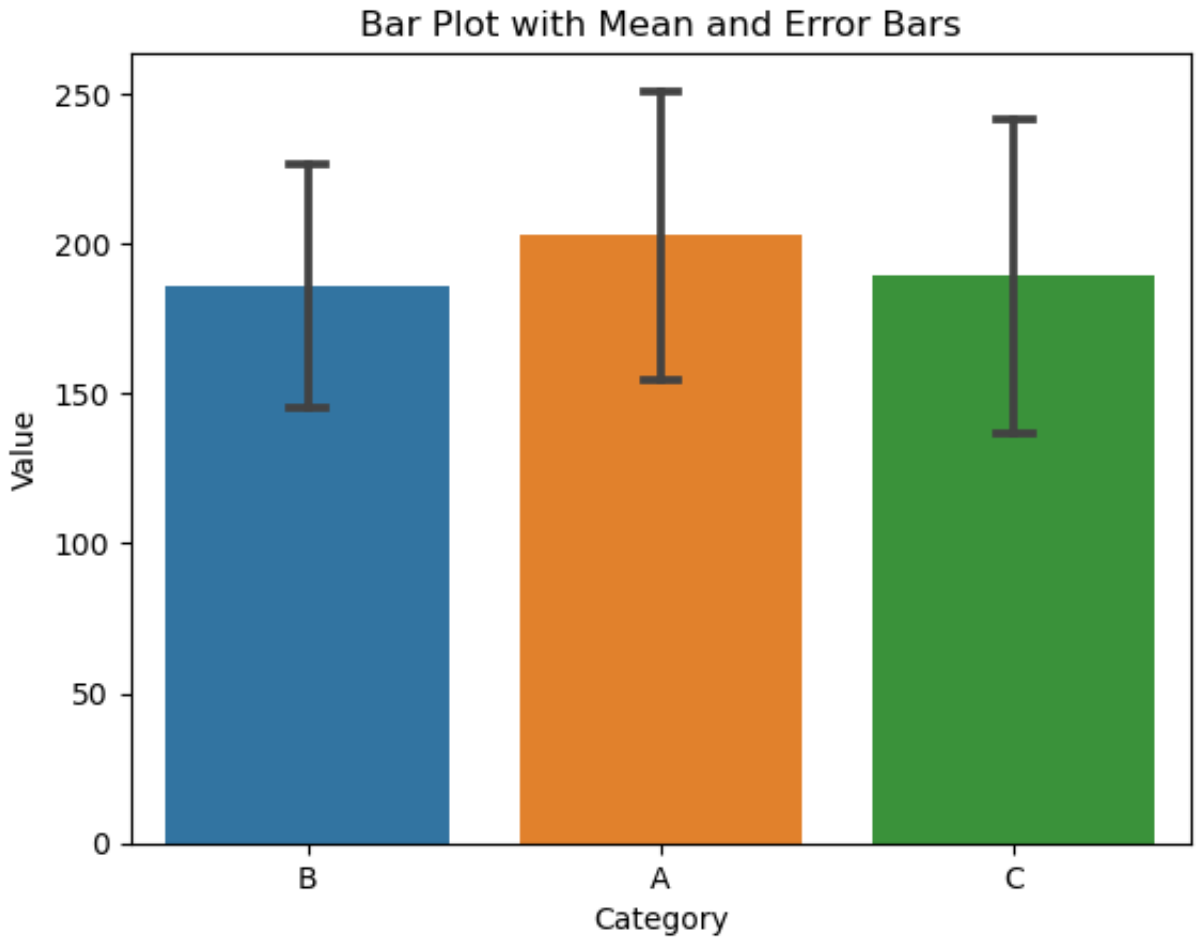
In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# 샘플 데이터 생성
data = {'Category': ['A', 'B', 'C', 'D'],
        'Value': [10, 20, 15, 5]}
df = pd.DataFrame(data)

# barplot 그리기
sns.barplot(x='Category', y='Value', data=df)

# 제목 추가
plt.title('Bar Plot Example')

plt.show()
```



3-2. 데이터의 평균과 표준편차를 막대그래프로 표시하기

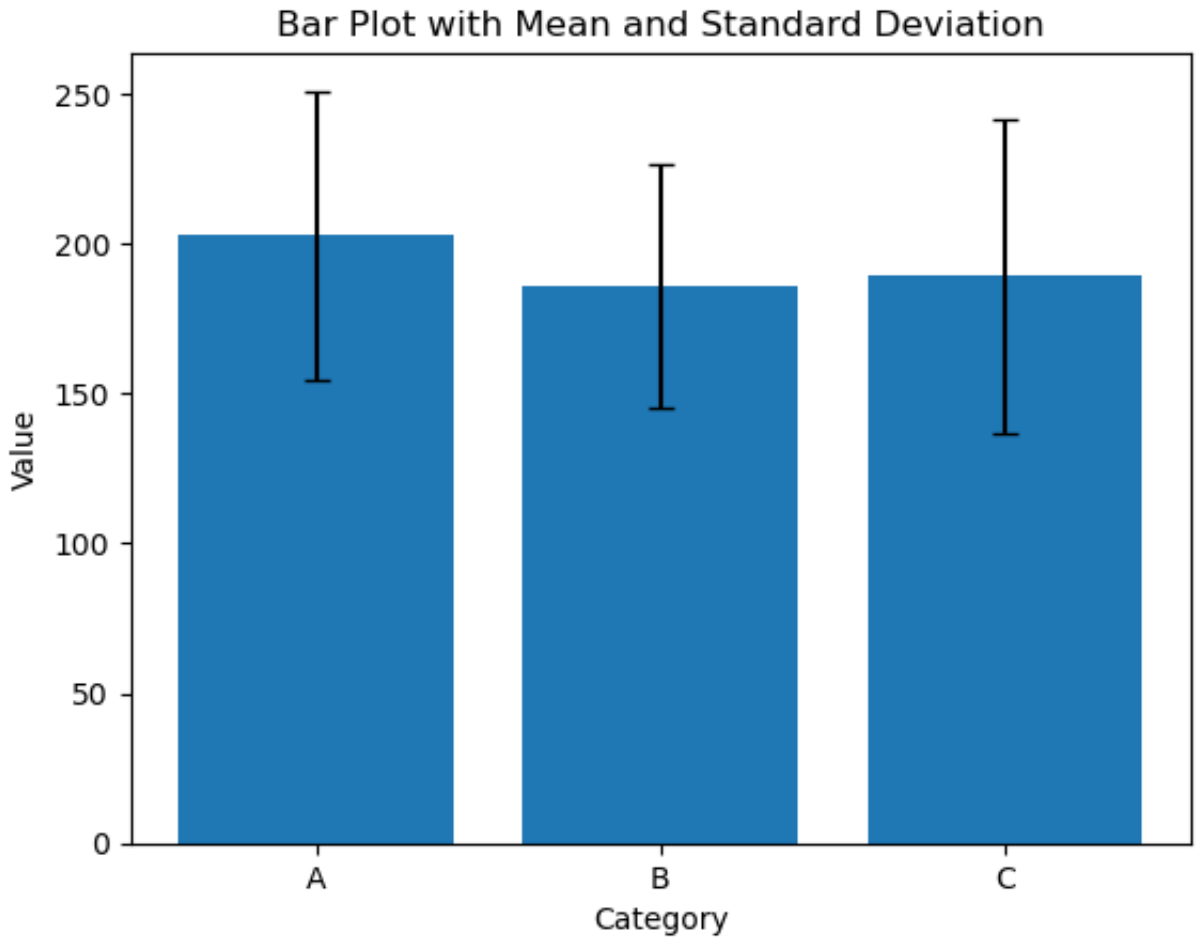
Seaborn을 사용하면 barplot으로 주어진 데이터 셋의 평균과 error bar를 쉽게 그래프로 구현할 수 있음. 이를 Matplotlib 만으로 구현하기 위해서는 데이터의 평균과 표준편차를 직접 계산한 후, plt.bar로 막대 그래프를 그리고, plt.errorbar를 사용하여 에러바를 추가해야 합니다.

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# 샘플 데이터 생성
np.random.seed(10)
data = {
    'Category': np.random.choice(['A', 'B', 'C'], 100),
    'Value': np.random.randn(100) * 50 + 200
}
df = pd.DataFrame(data)

# barplot 그리기
sns.barplot(
    x='Category', y='Value',
    data=df, errorbar='sd', capsize=0.1
)  # errorbar='sd'는 표준편차를 error bar로 표시

# 제목과 축 레이블 추가
plt.title('Bar Plot with Mean and Error Bars')
plt.xlabel('Category')
plt.ylabel('Value')

plt.show()
```



[Matplotlib 만으로 구현한 평균 및 표준편차 그래프]

In [ ]:
```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# 샘플 데이터 생성
np.random.seed(10)
data = {
    'Category': np.random.choice(['A', 'B', 'C'], 100),
    'Value': np.random.randn(100) * 50 + 200
}
df = pd.DataFrame(data)

# 범주별로 데이터의 평균과 표준편차 계산
means = df.groupby('Category')['Value'].mean()
stds = df.groupby('Category')['Value'].std()
categories = means.index

# 막대 그래프 그리기
fig, ax = plt.subplots()
bars = ax.bar(categories, means, yerr=stds, capsize=4)

# 제목과 축 레이블 추가
ax.set_title('Bar Plot with Mean and Standard Deviation')
ax.set_xlabel('Category')
ax.set_ylabel('Value')

plt.show()
```

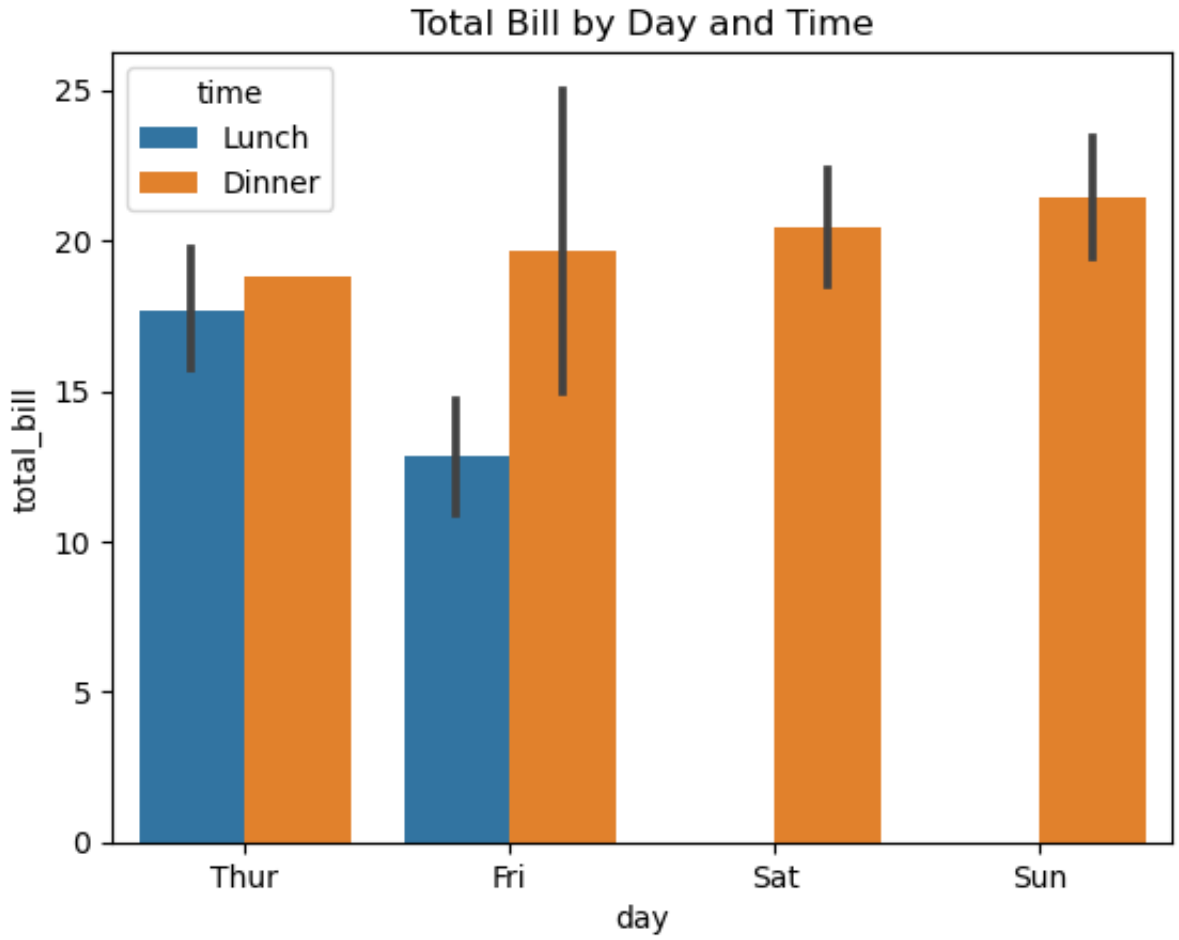


3-3. 여러 그룹의 데이터 비교하기

아래 예제에서는 Seaborn 라이브러리에서 제공하는 데이터셋('tips')을 사용합니다. 이 예제에서는 식사 요일('day')에 따른 총 청구 금액('total_bill')을 비교하고, 추가적으로 시간('time')에 따라 분류하여 비교합니다. 'hue='time'' 매개변수를 사용하여 점심과 저녁 식사('Lunch'와 'Dinner')에 따라 색상을 다르게 하여 데이터를 더 세분화하여 표시합니다.

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Seaborn에 내장된 팁 데이터셋 로드
tips = sns.load_dataset('tips')

# 막대 그래프 그리기
sns.barplot(x='day', y='total_bill', hue='time', data=tips)

# 제목 추가
plt.title('Total Bill by Day and Time')

plt.show()
```

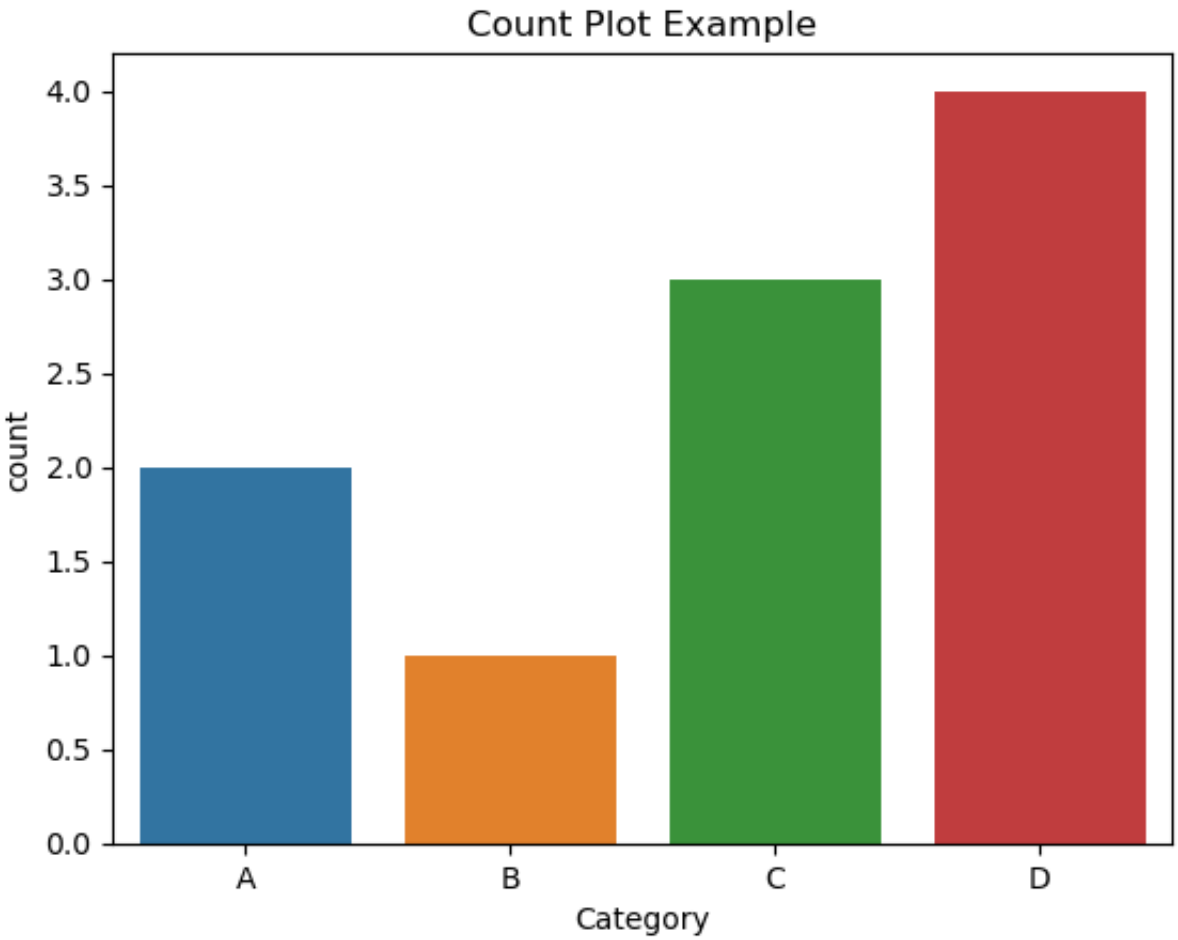
```
/Users/KP_Hong/anaconda3/envs/fastcampus/lib/python3.11/site-packages/seaborn/categorical.py:641: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```



3-4. countplot 사용하기

countplot은 데이터프레임 내의 범주형 컬럼을 선택하여, 각 범주에 속하는 데이터의 개수를 막대로 표현합니다. 이 함수는 barplot과 달리 y축의 수치 데이터를 필요로 하지 않으며, 데이터프레임 내에서 해당 범주의 빈도수를 직접 계산하여 그래프를 그립니다.

In [ ]:
```python
# 샘플 데이터 생성
data = {'Category': ['A', 'A', 'B', 'C', 'C', 'C', 'D', 'D', 'D', 'D']}
df = pd.DataFrame(data)

# countplot 그리기
sns.countplot(x='Category', data=df)

# 제목 추가
plt.title('Count Plot Example')

plt.show()
```



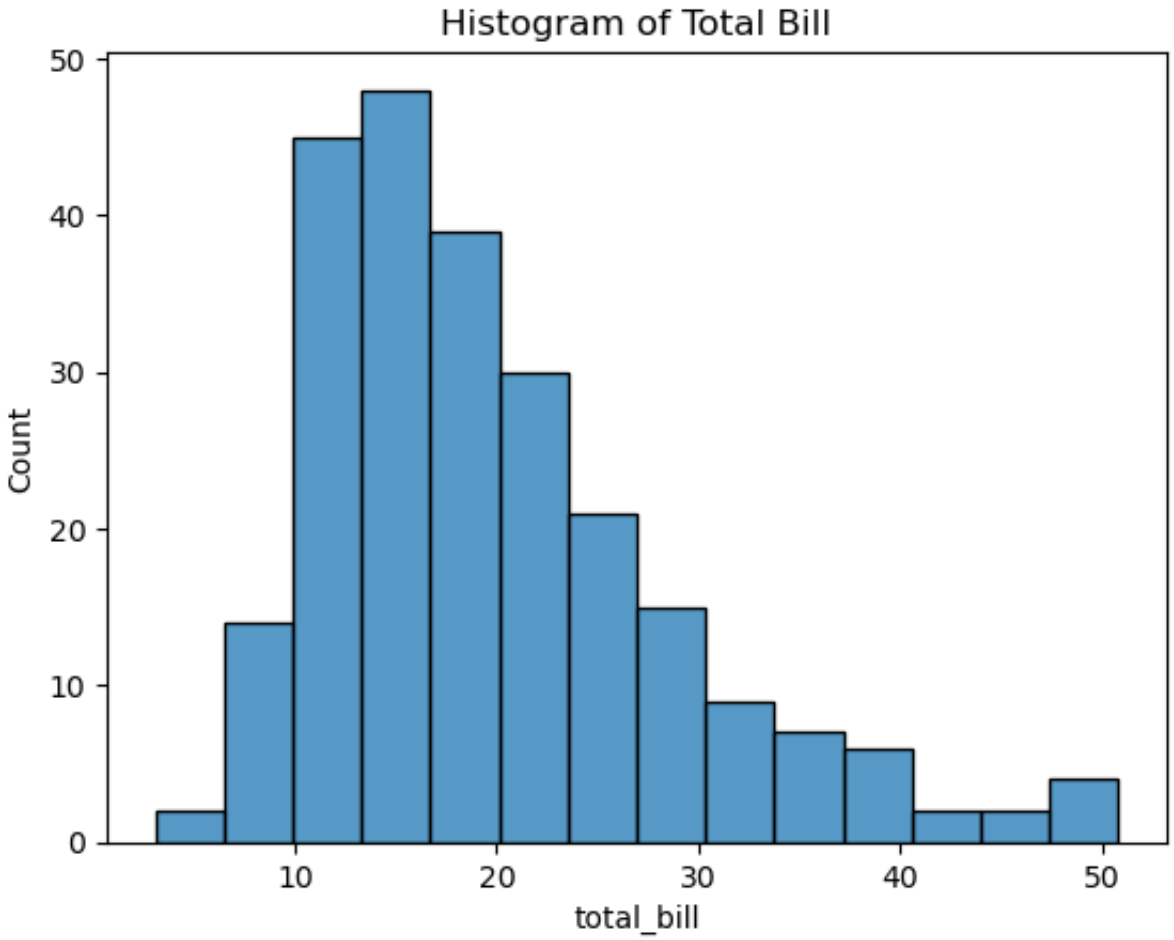
4. 히스토그램 그리기

histplot 메서드는 데이터의 분포를 시각화하는 데 사용되며, 데이터셋 내에서 변수의 빈도수를 막대 형태로 표현합니다. 이 메서드는 Seaborn 0.11.0 이상 버전에서 사용할 수 있으며, 이전 버전에서는 distplot을 사용했습니다.

4-1. 기본 히스토그램 그리기

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# 샘플 데이터 생성
data = sns.load_dataset("tips")  # Seaborn 내장 데이터셋 사용

# total_bill 컬럼에 대한 히스토그램 그리기
sns.histplot(data=data, x="total_bill")

# 그래프 제목 설정
plt.title('Histogram of Total Bill')

plt.show()
```

```
/Users/KP_Hong/anaconda3/envs/fastcampus/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

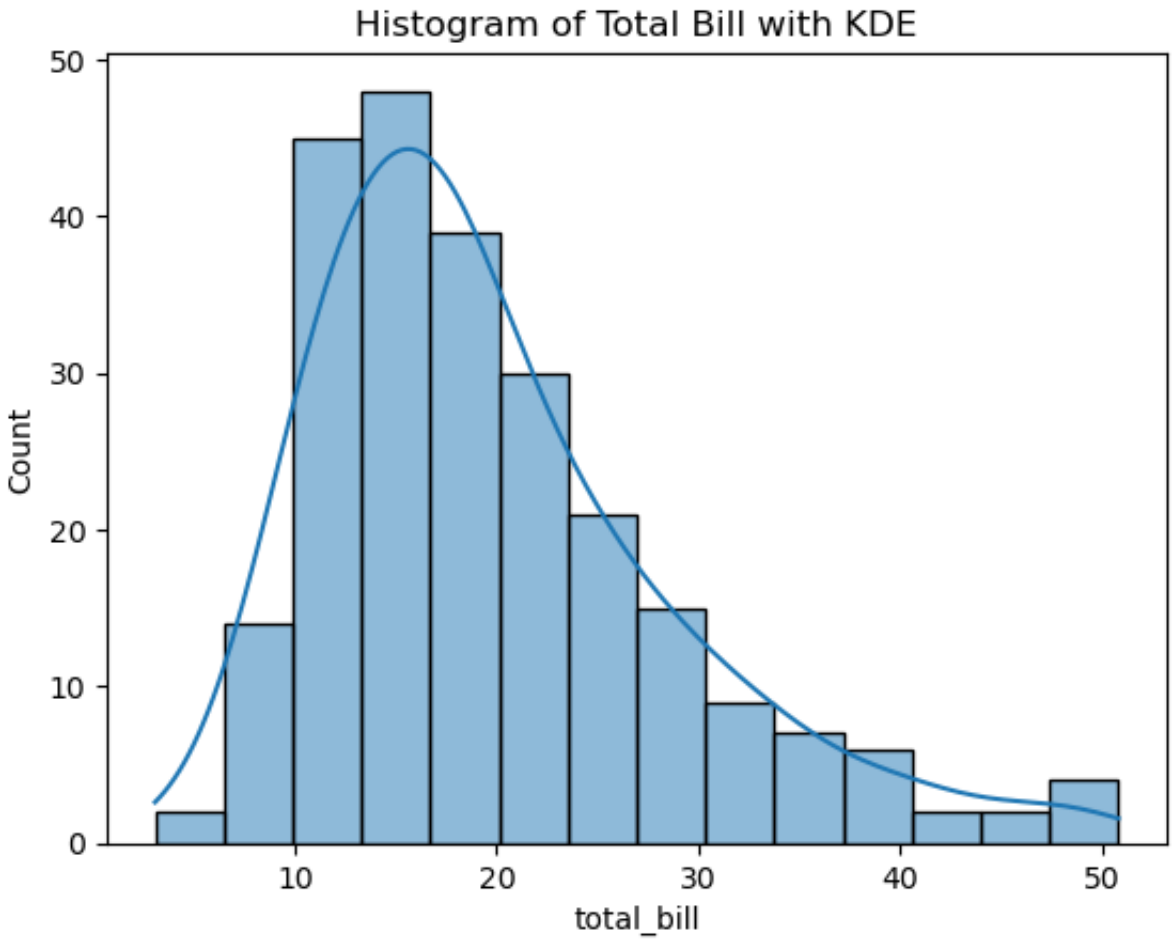


4-2. 히스토그램에 KDE(Kernel Density Estimate) 추가하기

히스토그램에 KDE(커널 밀도 추정) 곡선을 추가하여 데이터 분포를 더 부드럽게 표현할 수 있습니다.

In [ ]:
```
sns.histplot(data=data, x="total_bill", kde=True)
plt.title('Histogram of Total Bill with KDE')
plt.show()
```

```
/Users/KP_Hong/anaconda3/envs/fastcampus/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

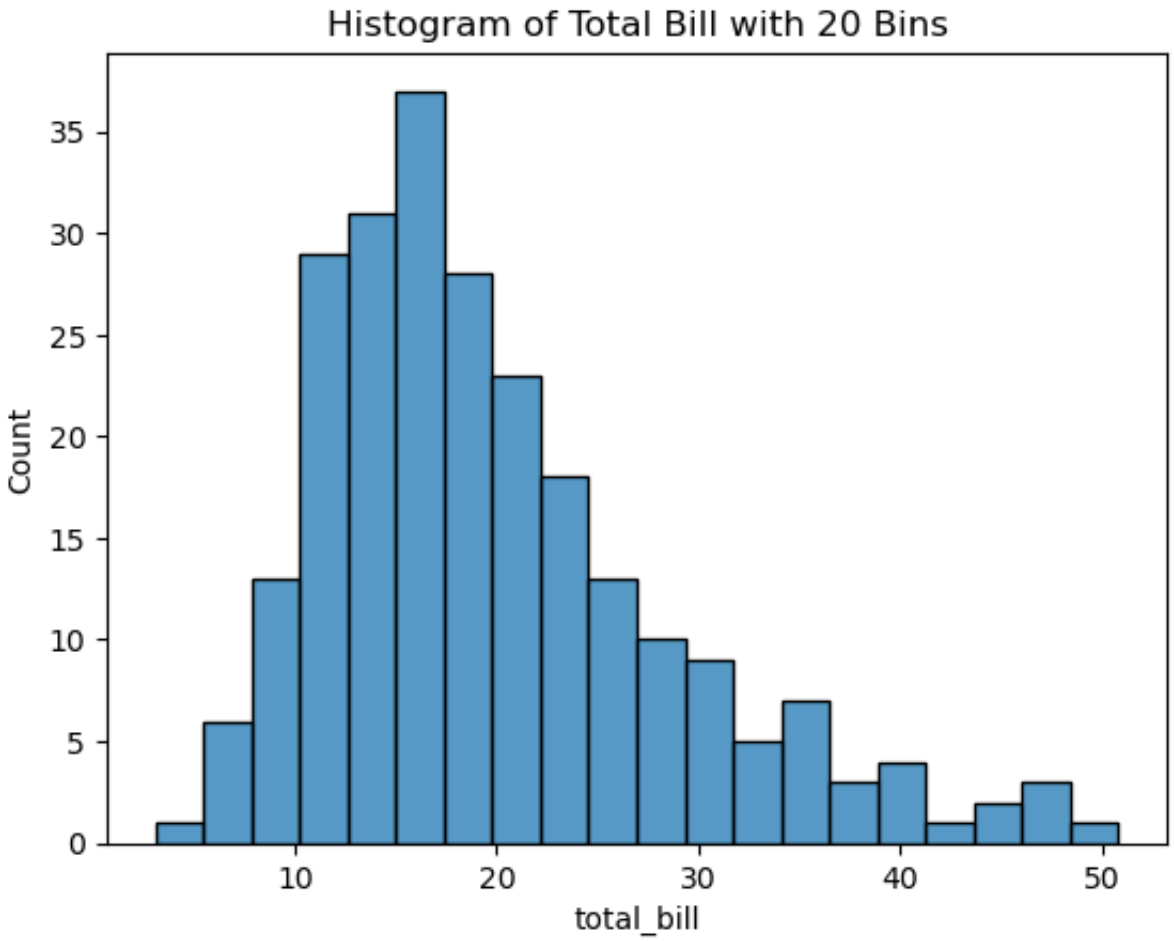


4-3. 빈(bin)의 수 조정하기

히스토그램의 빈(bin)의 수를 조정하여 데이터의 분포를 더 세밀하게 또는 더 굵게 표현할 수 있습니다.

In [ ]:
```
sns.histplot(data=data, x="total_bill", bins=20)
plt.title('Histogram of Total Bill with 20 Bins')
plt.show()
```

```
/Users/KP_Hong/anaconda3/envs/fastcampus/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

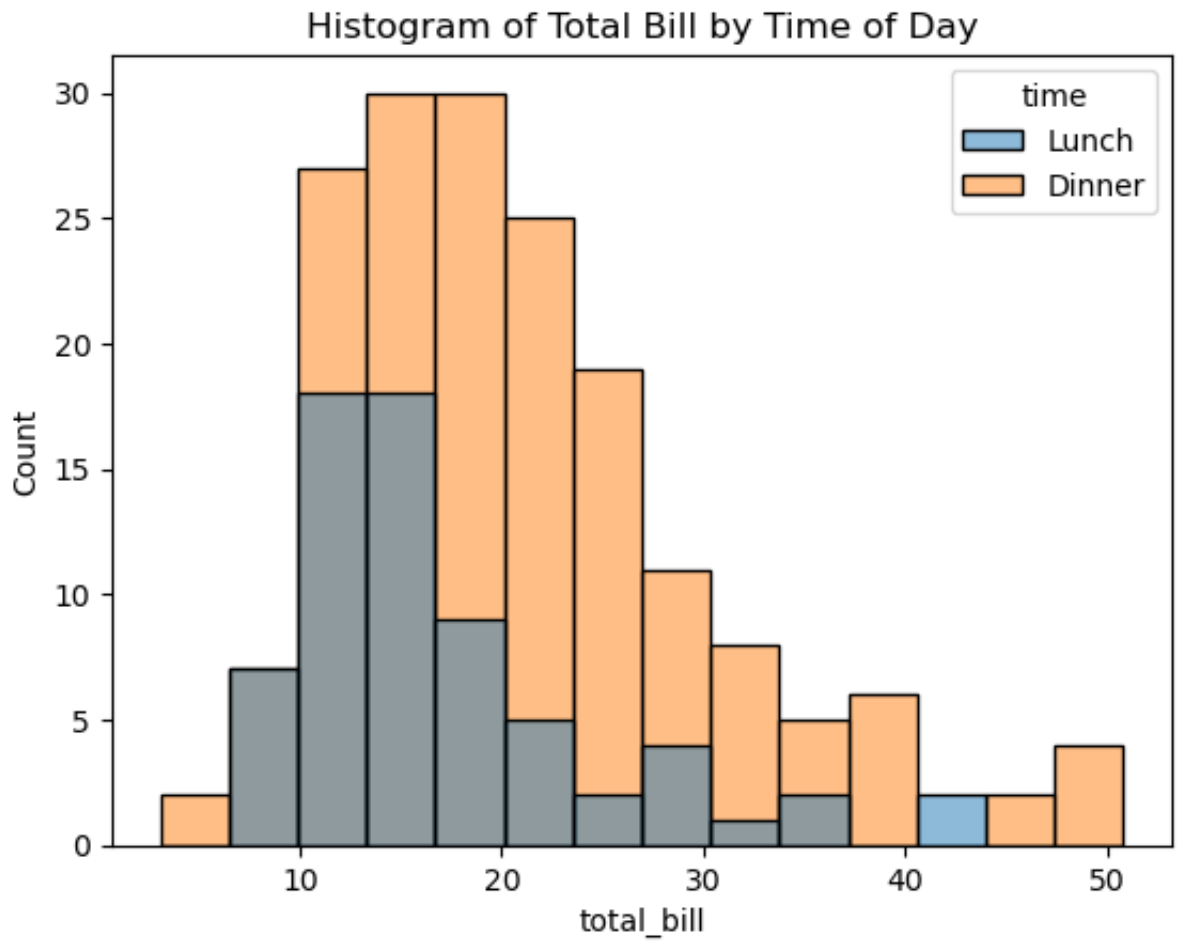


4-4. 카테고리별 히스토그램 그리기

hue 매개변수를 사용하여 카테고리별로 히스토그램을 그리고 색상을 다르게 할 수 있습니다.

In [ ]:
```
sns.histplot(data=data, x="total_bill", hue="time")
plt.title('Histogram of Total Bill by Time of Day')
plt.show()
```

```
/Users/KP_Hong/anaconda3/envs/fastcampus/lib/python3.11/site-packages/sea
born/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated
and will be removed in a future version. Convert inf values to NaN before
operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/Users/KP_Hong/anaconda3/envs/fastcampus/lib/python3.11/site-packages/sea
born/_oldcore.py:1057: FutureWarning: The default of observed=False is de
precated and will be changed to True in a future version of pandas. Pass
observed=False to retain current behavior or observed=True to adopt the f
uture default and silence this warning.
  grouped_data = data.groupby(
```



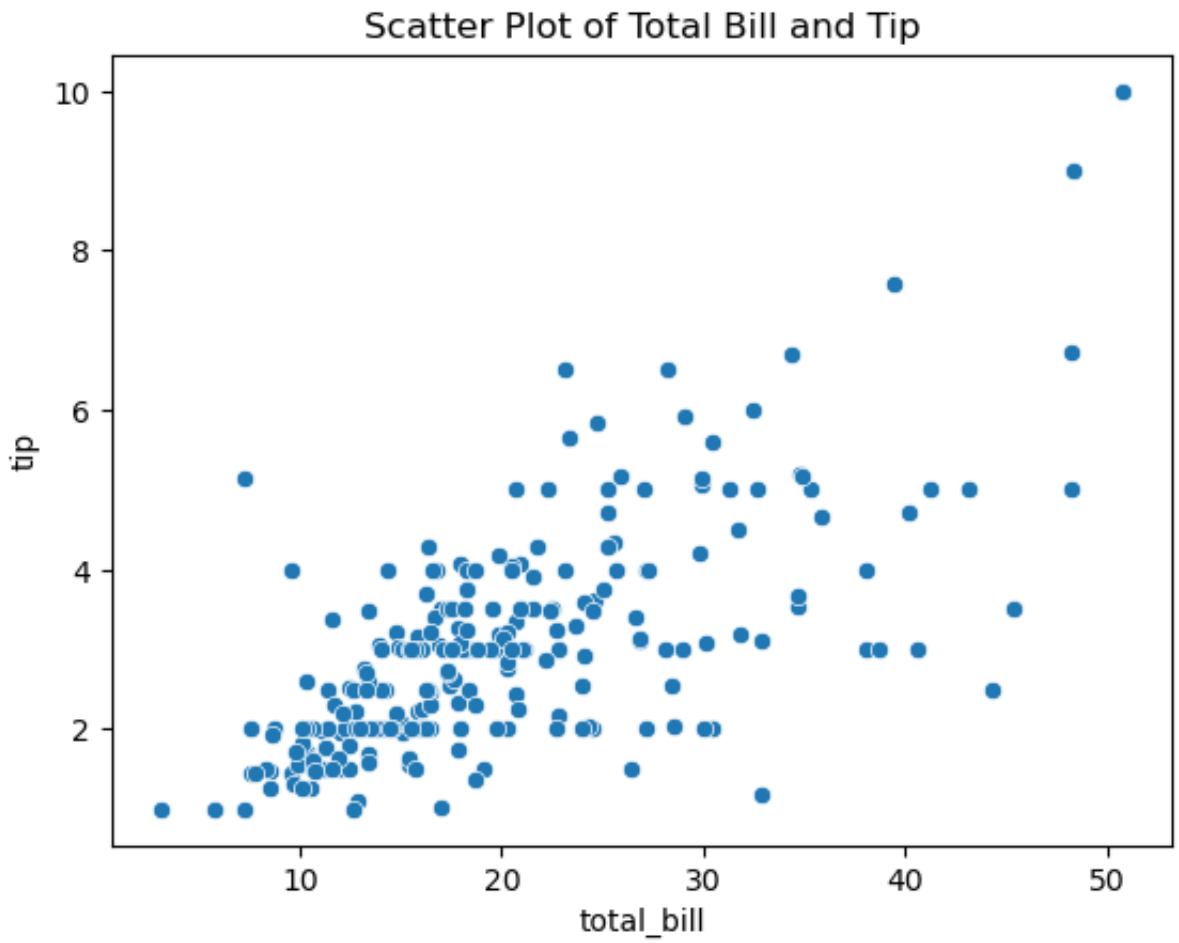
5. 산점도 그리기

scatterplot 메서드를 사용하여 산점도(scatter plot)를 그릴 수 있습니다. Seaborn의 scatterplot은 두 수치형 변수 간의 관계를 점으로 표현해주며, 추가적으로 범주형 변수에 따른 색상(hue)이나 크기(size)를 다르게 하여 더 많은 정보를 제공할 수 있습니다.

5-1. 기본 산점도 그리기

아래 예시는 Seaborn에 내장된 tips 데이터셋을 사용하여, 총 청구 금액(total_bill)과 팁(tip) 간의 관계를 산점도로 나타냅니다.

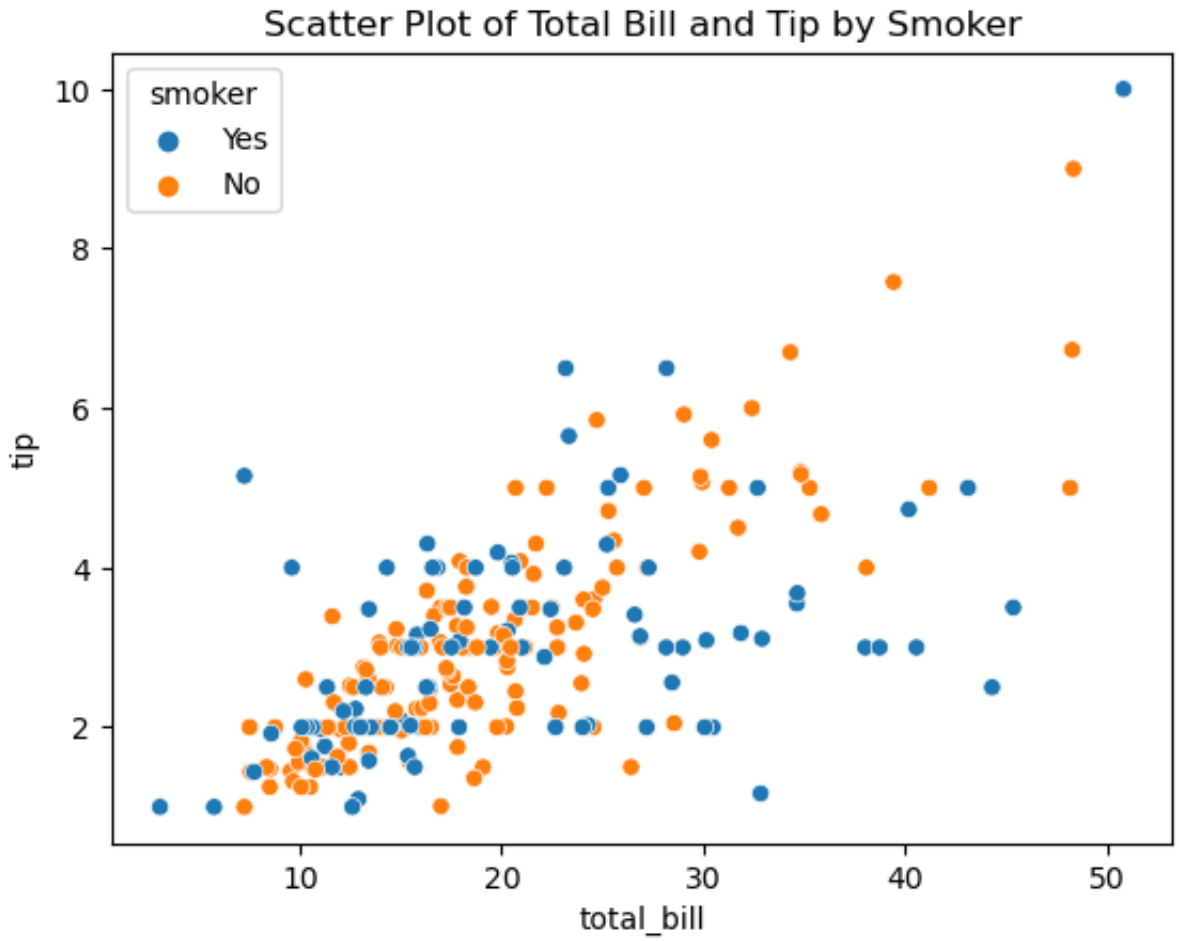
In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# 샘플 데이터 로드
tips = sns.load_dataset('tips')

# 기본 산점도 그리기
sns.scatterplot(x='total_bill', y='tip', data=tips)

plt.title('Scatter Plot of Total Bill and Tip')
plt.show()
```



5-2. 범주형 변수에 따른 색상 구분

hue 매개변수를 사용하여 데이터 포인트를 범주형 변수에 따라 색상을 다르게 할 수 있습니다.

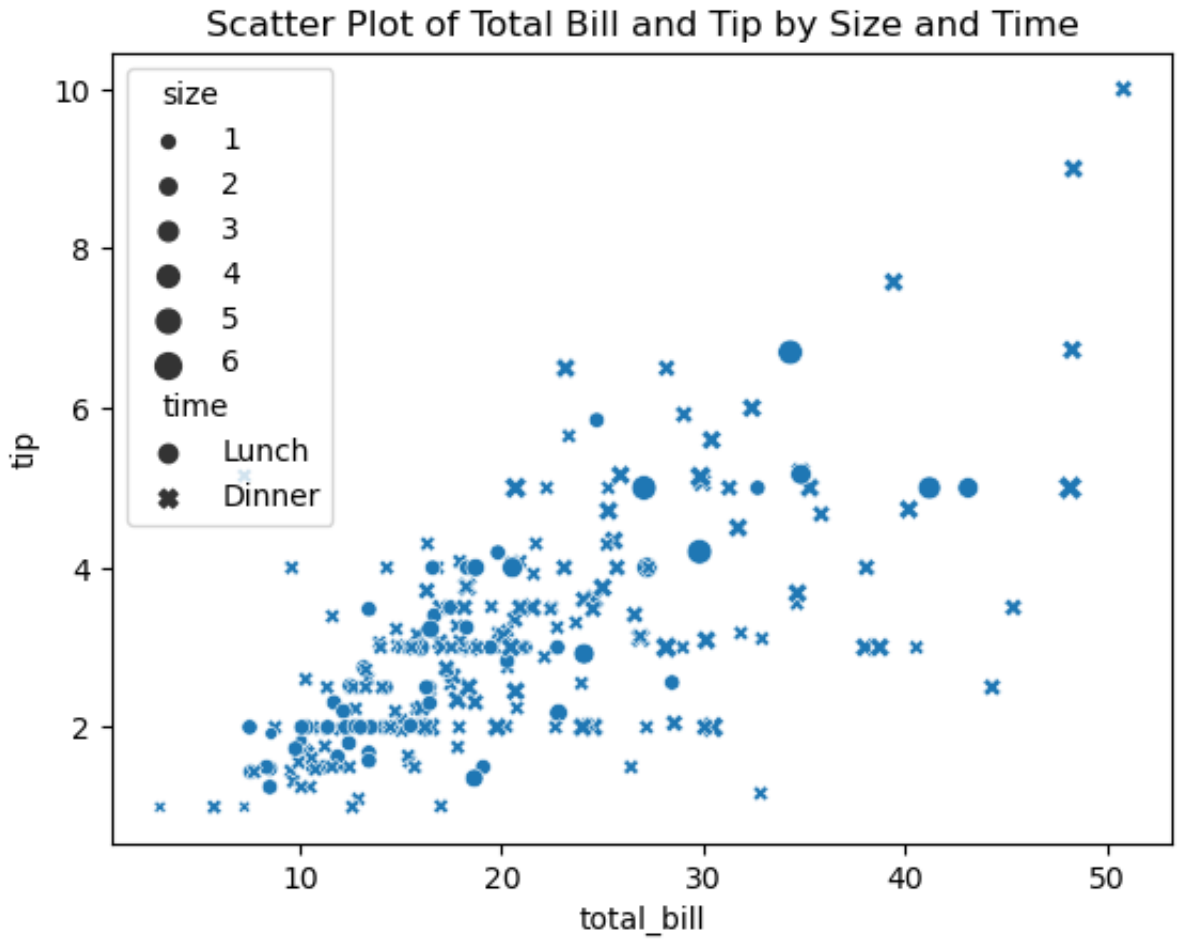
In [ ]:
```
sns.scatterplot(x='total_bill', y='tip', hue='smoker', data=tips)

plt.title('Scatter Plot of Total Bill and Tip by Smoker')
plt.show()
```



5-3. 점의 크기와 스타일 조정

size와 style 매개변수를 사용하여 점의 크기와 스타일을 각각 다르게 설정할 수 있습니다.

In [ ]:
```
sns.scatterplot(
    x='total_bill', y='tip',
    size='size', style='time', data=tips
)

plt.title('Scatter Plot of Total Bill and Tip by Size and Time')
plt.show()
```


Scatter Plot of Total Bill and Tip by Size and Time
size
1
2
3
4
5
6
time
Lunch
Dinner
tip
total_bill
10
8
6
4
2
10
20
30
40
50