

3주차-2

1. 오늘은 무엇을 배우는가?

2. Seaborn

2.1. 패키지 불러오기

2.2. 데이터 불러오기

2.3. 그래프 - catplot

2.3.1. stripplot

2.3.2. swarmplot

2.3.3. boxplot

2.3.4. 막대 그래프

2.3.5 히스토그램(도수분포표)

3. folium

3.1. folium이란?

3.2. 패키지 불러오기

3.3. 위치 설정, 마커 생성하기

3.4. 원 생성하기

1. 오늘은 무엇을 배우는가?

1. 패키지 불러오기

2. 데이터 불러오기

3. 그래프 - catplot

a. stripplot

b. swarmplot

c. boxplot

d. 막대 그래프

e. 히스토그램(도수분포표)

4. folium

a. 마커 추가하기

b. 원 추가하기

2. Seaborn

2.1. 패키지 불러오기

- 데이터를 다루는 데 필요한 패키지를 불러옵니다.

```
import pandas as pd
import seaborn as sns
pd.options.display.float_format = '{:.2f}'.format
```

2.2. 데이터 불러오기

- 3주차-1 과제에서 사용한 팀 데이터를 사용합니다.

```
팀_데이터 = pd.read_excel("팀.xlsx")
팀_데이터
```

2.3. 그래프 - catplot

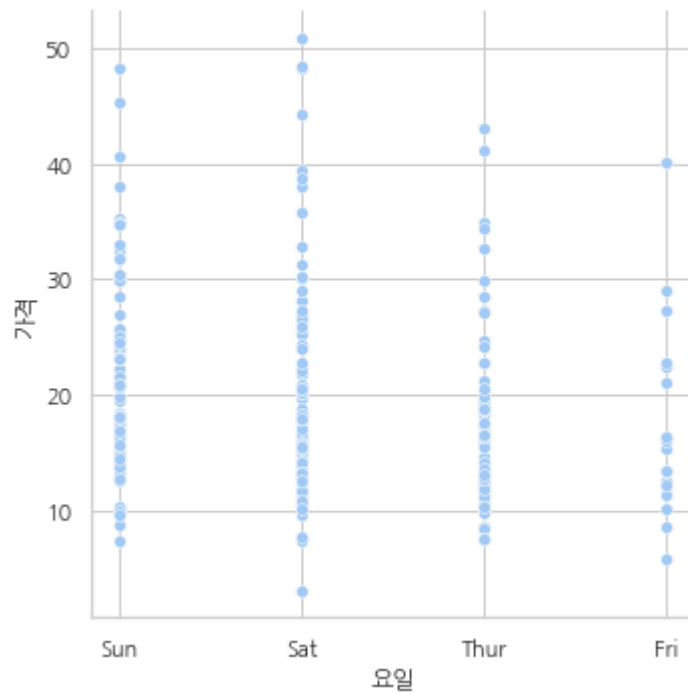
- catplot은 두 축의 데이터 중 한 축의 데이터가 범주형일 때 사용하는 함수입니다.

2.3.1. stripplot

- stripplot은 한 축이 범주형 데이터인 scatterplot을 의미합니다.
catplot의 기본값이 stripplot의 형태이므로, catplot 함수를 사용합니다.
일반적인 scatterplot은 한 줄로 데이터가 세워집니다.

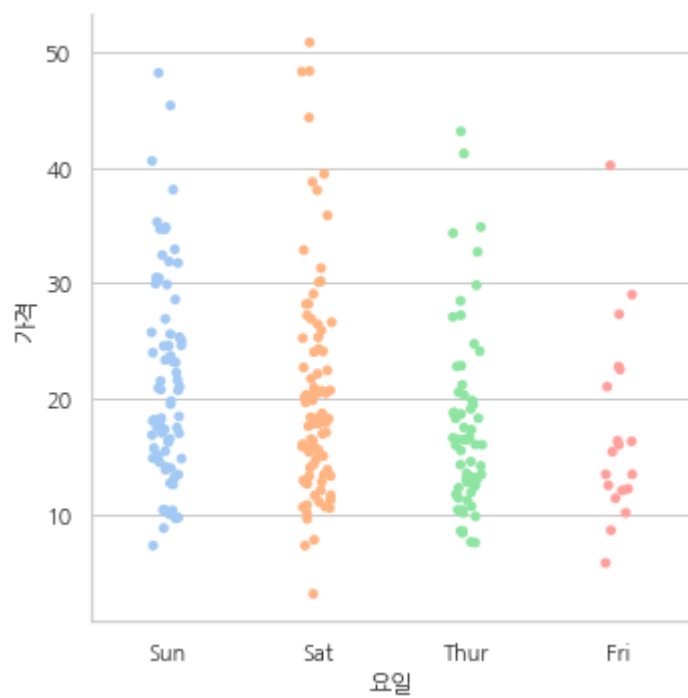
```
X축 = "요일"
Y축 = "가격"
데이터 = 팀_데이터

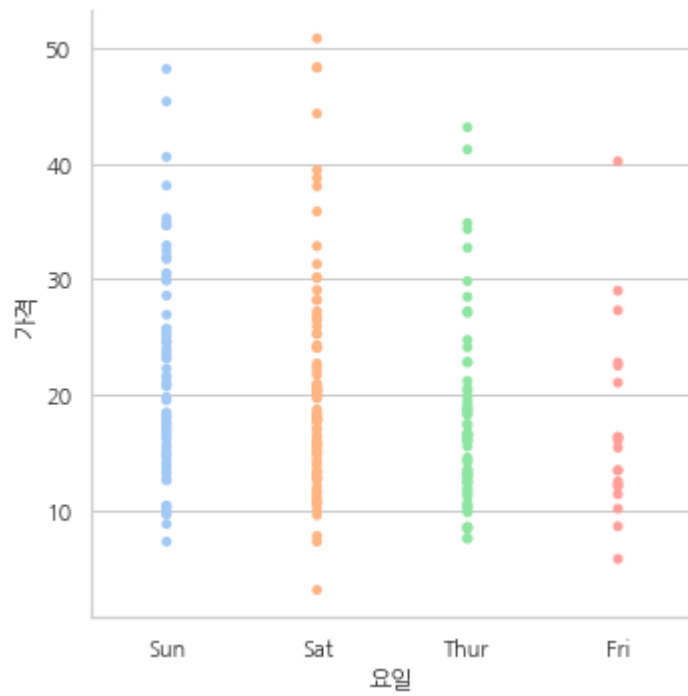
sns.relplot(x=X축, y=Y축, data=데이터) # 일반 scatterplot은 데이터를 한 줄로 세워줍니다.
```



- catplot을 사용하여 stripplot을 출력할 수 있습니다.
jitter=False로 파라미터를 설정하면 일반 scatterplot의 형태가 됩니다.

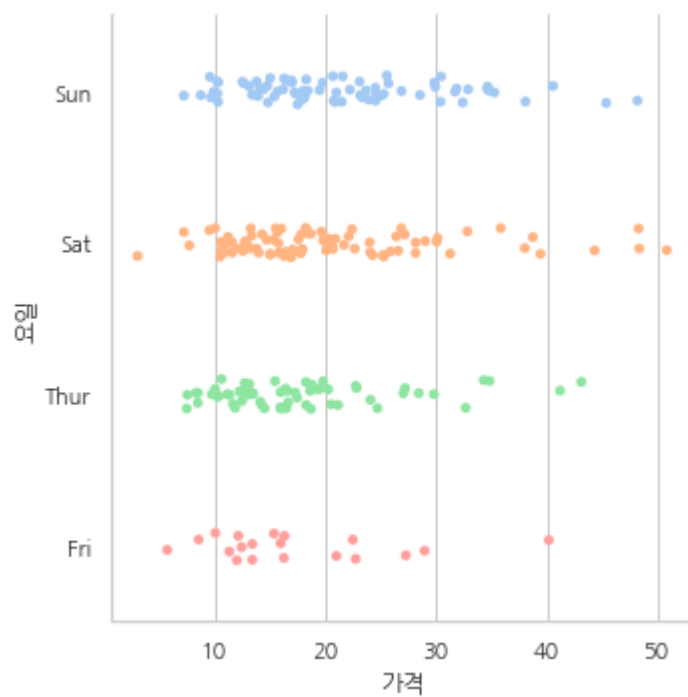
```
sns.catplot(x=X축, y=Y축, data=데이터)
sns.catplot(x=X축, y=Y축, data=데이터, jitter=False)
```





- 축에 들어갈 데이터를 바꿈으로써 그래프를 회전한 효과를 낼 수 있습니다.

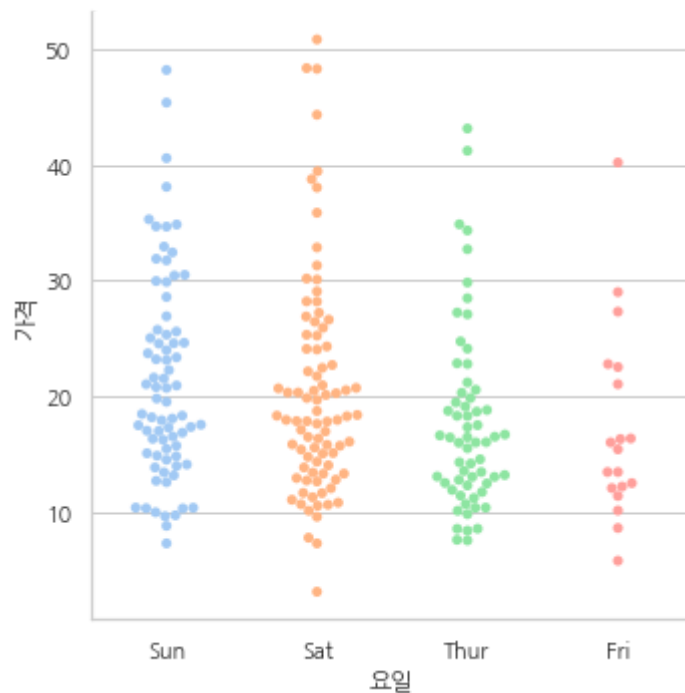
```
x축 = "가격" # X, Y축을 반대로 입력합니다.
y축 = "요일"
데이터 = 팁_데이터
sns.catplot(x=x축, y=y축, data=데이터)
```



2.3.2. swarmplot

- stripplot과 비슷하지만 그래프의 점들이 겹쳐지지 않습니다.
파라미터 kind="swarm"을 제공하여 출력할 수 있습니다.

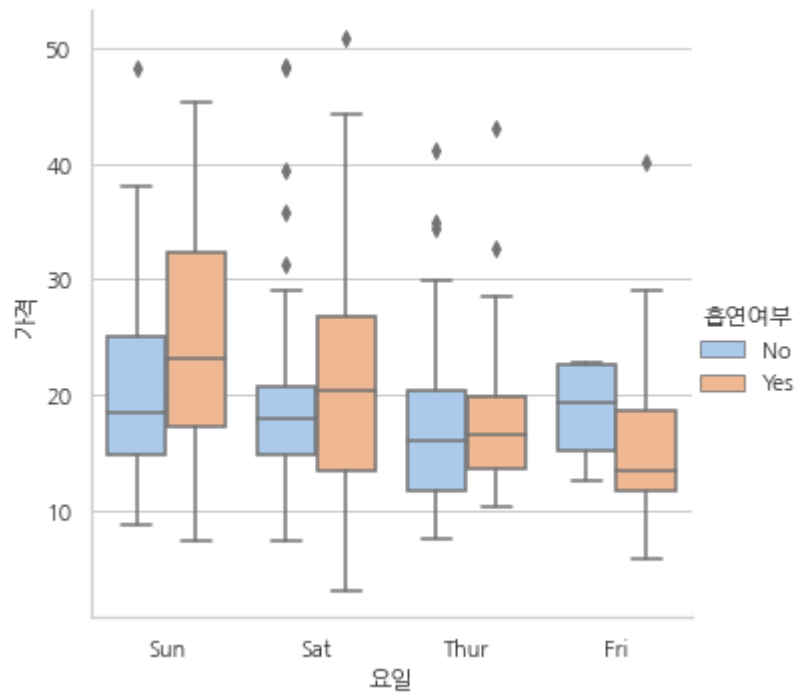
```
sns.catplot(x=X축, y=Y축, data=데이터, kind="swarm")
```



2.3.3. boxplot

- boxplot은 위에서 언급한 stripplot, swarmplot과 달리 분포를 나타내는 차트입니다.
- 맨 윗부분은 outlier를 제외한 최댓값, 맨 아랫부분은 outlier를 제외한 최솟값입니다.
- 가운데 상자에서 윗부분은 데이터의 25% 지점(1분위수, Q1), 중간 부분은 50% 지점(2분위수, Q2, 평균이 아니라 중앙값), 아랫부분은 75% 지점(3분위수, Q3)입니다.
- 파라미터 kind="box"를 제공하여 출력할 수 있습니다.

```
sns.catplot(x=X축, y=Y축, data=데이터, kind="box", hue=색구분기준)
```



2.3.4. 막대 그래프

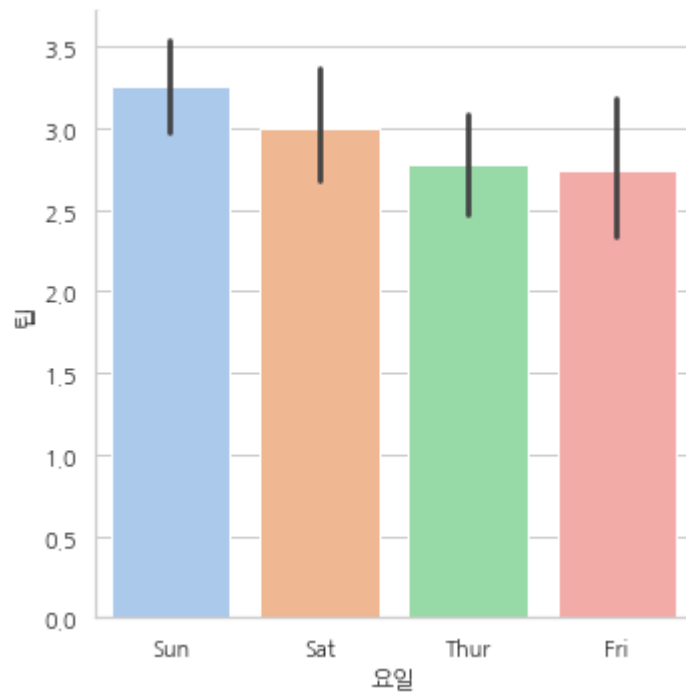
- 한 축은 범주형 데이터가 들어가며, 다른 한 축은 데이터의 평균값이 들어갑니다. 데이터 개수가 아닙니다!
- 파라미터 kind="bar"을 제공하여 출력할 수 있습니다.

```

X축 = "요일"
Y축 = "팁"
데이터 = 팁_데이터

sns.catplot(x=X축, y=Y축, data=데이터, kind="bar")

```

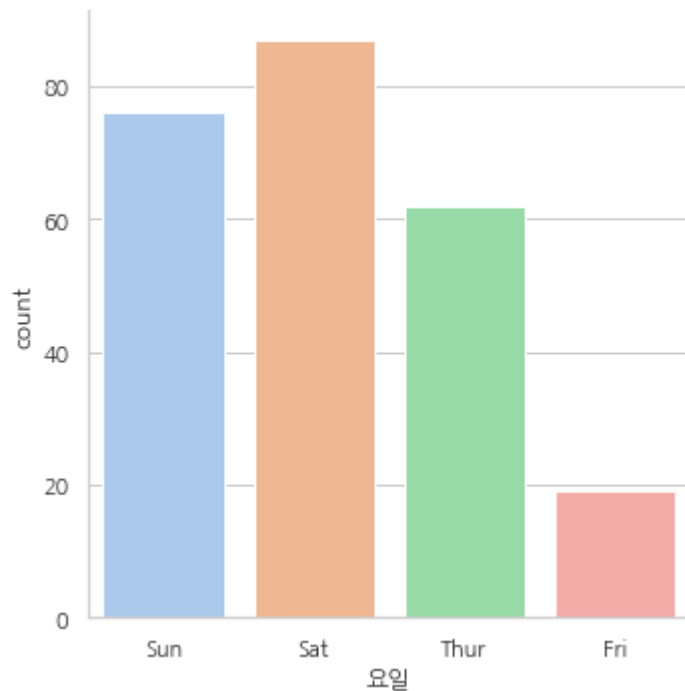


2.3.5 히스토그램(도수분포표)

- 특정 데이터의 개수를 막대 그래프로 나타낸 그래프를 히스토그램이라고 합니다.
- 파라미터 kind="bar"을 제공하여 출력할 수 있습니다.
- 참고 : 히스토그램은 y축 데이터가 필요하지 않습니다.

```
x축 = "요일"

sns.catplot(x=x축, data=데이터, kind="count")
```



3. folium

3.1. folium이란?

- 파이썬을 통한 지도 데이터 시각화 라이브러리입니다.

3.2. 패키지 불러오기

```
import folium
```

3.3. 위치 설정, 마커 생성하기

- folium.map()으로 지도를 생성합니다.
- folium.marker()로 마커를 설정하고, add_to() 함수를 통해 지도에 추가할 수 있습니다.

```
창원광장 = [35.226367, 128.682232]
m = folium.Map(location=창원광장, zoom_start=17)
m.add_child(folium.ClickForMarker())

내용 = "창원광장입니다."
툴팁 = "창원광장"
folium.Marker(location=창원광장, popup=내용, tooltip=툴팁).add_to(m)

m
```




3.4. 원 생성하기

- 특정 지점을 중심으로 원을 표기할 수 있습니다.

```
창원시청 = [35.228356, 128.682073]
```

```
내용 = "창원시청입니다"
```

```
툴팁 = "창원시청"
```

```
구름_아이콘 = folium.Icon(icon="cloud")
```

```
folium.Marker(location=창원시청, popup=내용, tooltip=툴팁, icon=구름_아이콘).add_to(m)
```

```
원_반지름 = 100
```

```
내용 = "창원시청의 범위는 이정도"
```

```
파란색 = "#3186cc"
```

```
folium.CircleMarker(location=창원시청, radius=원_반지름, popup=내용, color=파란색, fill=True, fill_color=파란색).add_to(m)
```

```
m
```

