



CNU 데이터 분석 교육



6th lecture
"Visualization2"

2021 - 11 - 02

지난시간?

1. `matplotlib` – `plot`, `scatter`

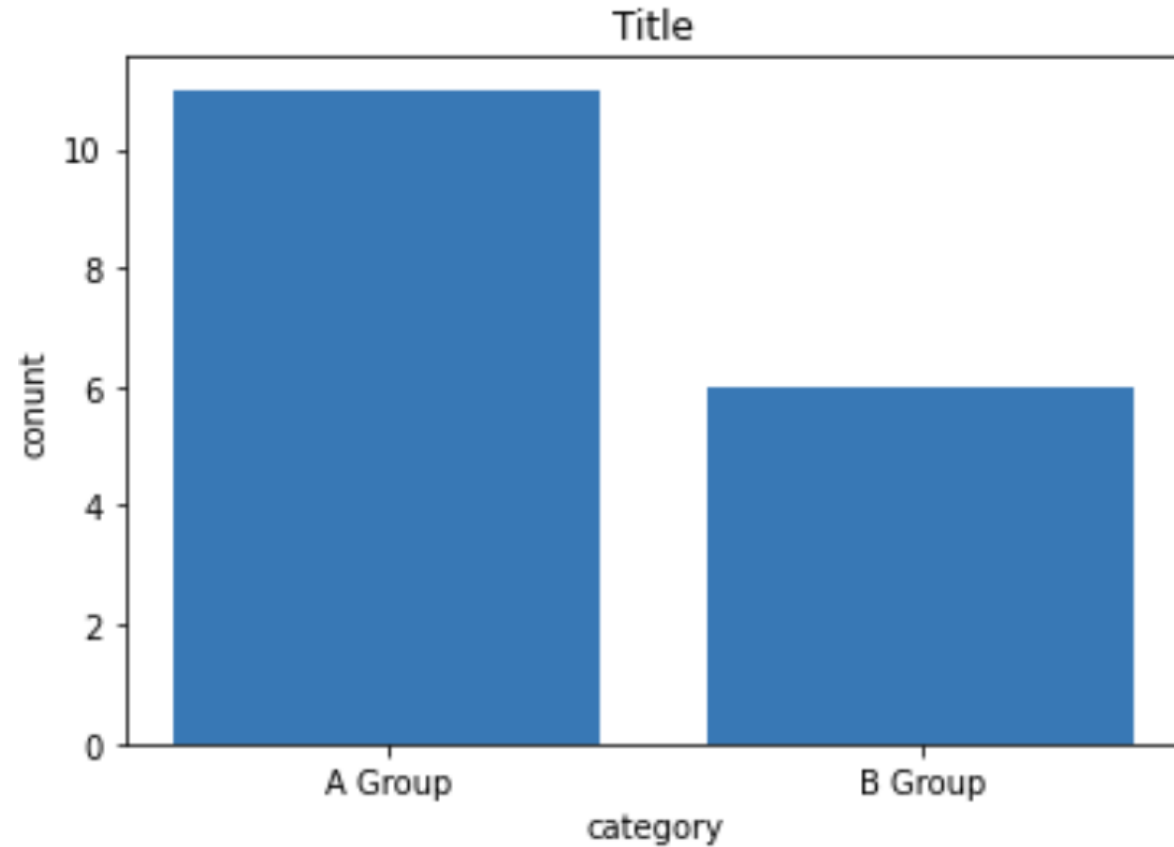
2. `Seaborn` – `heatmap`, `scatter`, `box`

오늘은 무엇을?

1. Seaborn - barplot, pie chart, displot

barplot

1. 범주별로 자료의 개수를 세어 막대로 나타내는 시각화 기법
2. 자료의 크기정도를 높이를 통해 비교 요약하여 나타내준다.



barplot

seaborn.barplot

`seaborn.barplot(*, x=None, y=None, hue=None, data=None, order=None, hue_order=None, estimator=<function mean at 0x7ff320f315e0>, ci=95, n_boot=1000, units=None, seed=None, orient=None, color=None, palette=None, saturation=0.75, errcolor='.26', errwidth=None, capsize=None, dodge=True, ax=None, **kwargs)`

Show point estimates and confidence intervals as rectangular bars.

A bar plot represents an estimate of central tendency for a numeric variable with the height of each rectangle and provides some indication of the uncertainty around that estimate using error bars. Bar plots include 0 in the quantitative axis range, and they are a good choice when 0 is a meaningful value for the quantitative variable, and you want to make comparisons against it.

For datasets where 0 is not a meaningful value, a point plot will allow you to focus on differences between levels of one or more categorical variables.

It is also important to keep in mind that a bar plot shows only the mean (or other estimator) value, but in many cases it may be more informative to show the distribution of values at each level of the categorical variables. In that case, other approaches such as a box or violin plot may be more appropriate.

Input data can be passed in a variety of formats, including:

- Vectors of data represented as lists, numpy arrays, or pandas Series objects passed directly to the `x`, `y`, and/or `hue` parameters.
- A "long-form" DataFrame, in which case the `x`, `y`, and `hue` variables will determine how the data are plotted.
- A "wide-form" DataFrame, such that each numeric column will be plotted.
- An array or list of vectors.

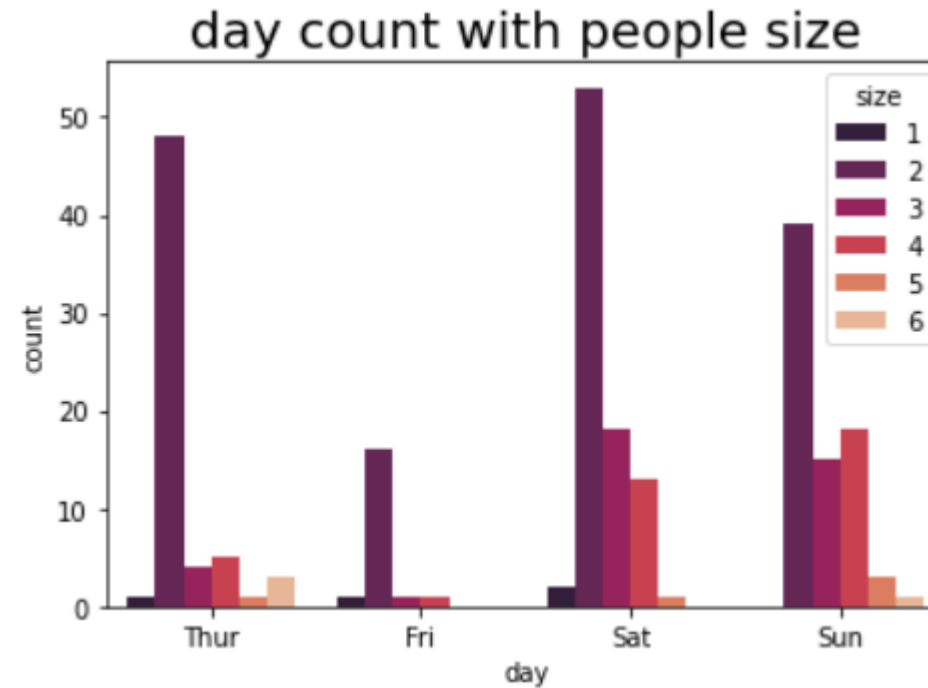
In most cases, it is possible to use numpy or Python objects, but pandas objects are preferable because the associated names will be used to annotate the axes. Additionally, you can use Categorical types for the grouping variables to control the order of plot elements.

This function always treats one of the variables as categorical and draws data at ordinal positions (0, 1, ... n) on the relevant axis, even when the data has a numeric or date type.

See the [tutorial](#) for more information.

문제 1

요일별 식당 방문횟수 (*count*)를 식사인원별 (*size*)로 나눠서 시각화 하라,
그리고 분석하라.



문제 1

정답

```
# 문제 1
|
sns.countplot(x = 'day', data = tips, palette='rocket', hue = 'size', saturation = 0.8)

plt.title("day count with people size", fontsize = 20)
plt.show()
```

평소 2인 식사를 많이 한다.

주말에는 3인 이상의 사람들이 평일보다 더 많이 온다.

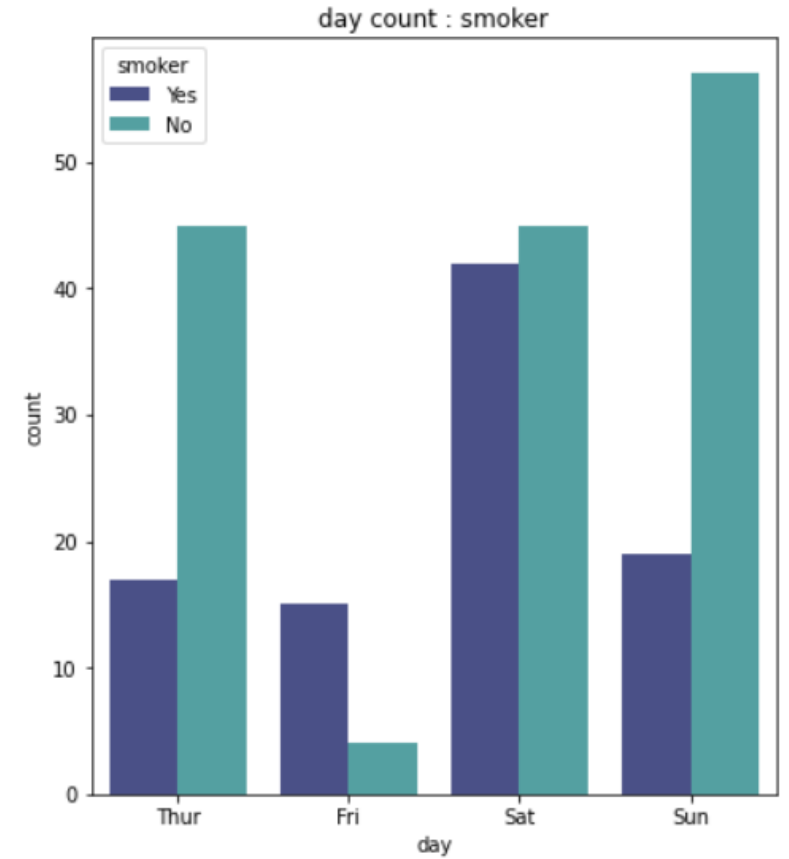
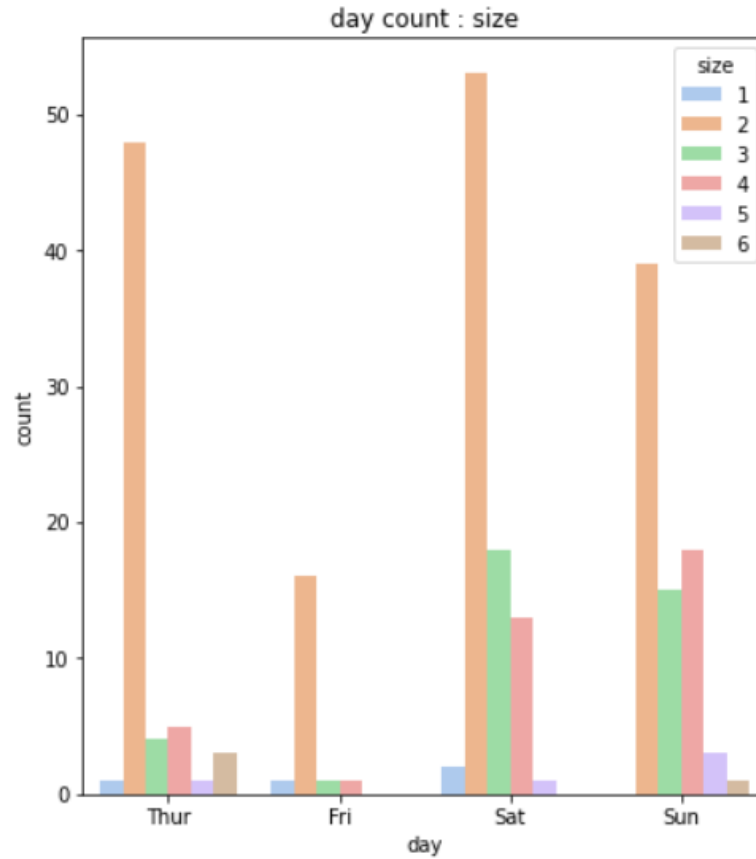
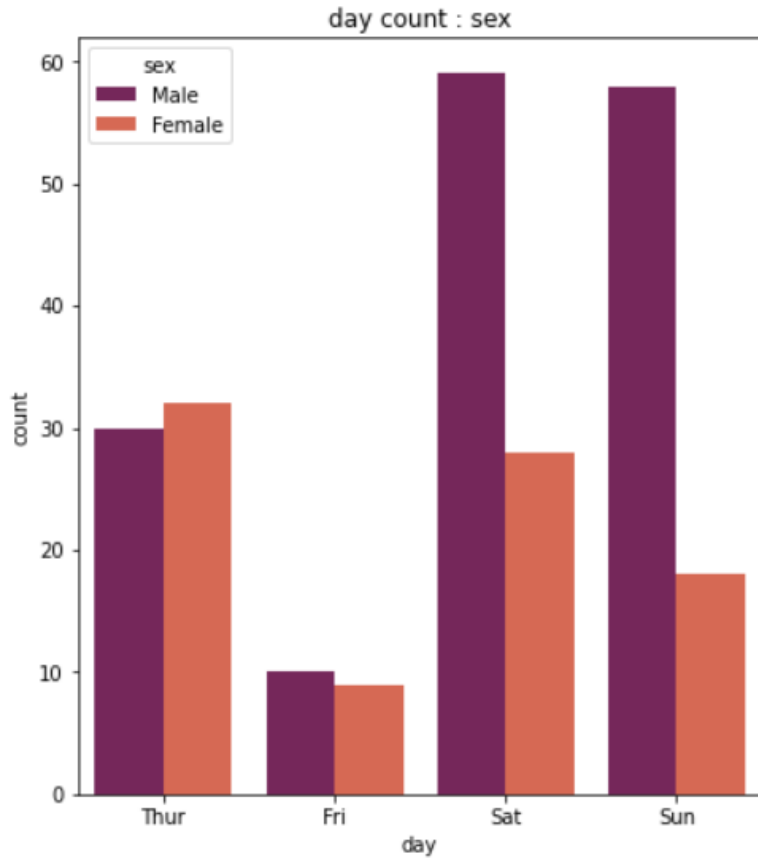
-> 주말에는 단체손님을 맞이할 준비를 평일보다 철저히 해야한다.

-> 요일에 따라 테이블의 배열, 재료 수급, 손질 등 식당 운영 전략의 차별화에 활용.

subplot

각각의 plot을 한꺼번에 나열하는 기능

day countplot



문제 2

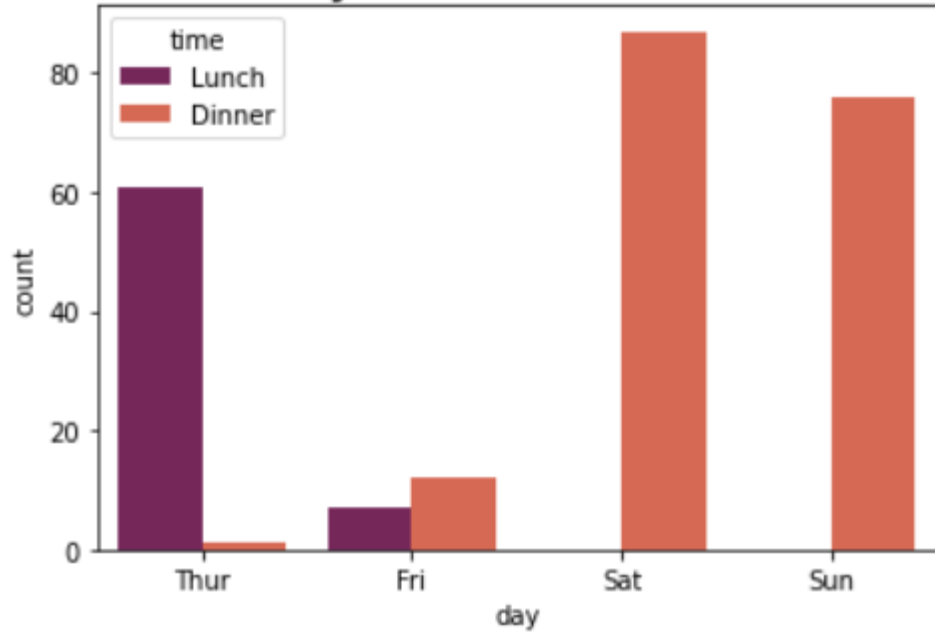
당신의 가게가 번창하여 새로운 알바생을 구하려고한다. 알바생이 일을 할 수 있는 시간은 하루 6시간, 최대 이틀이다. 이에 어느 시간에 알바생을 고용하는것이 가장 현명한 선택인가?

tips 데이터셋을 이용하여 다음을 시각화 하여라.

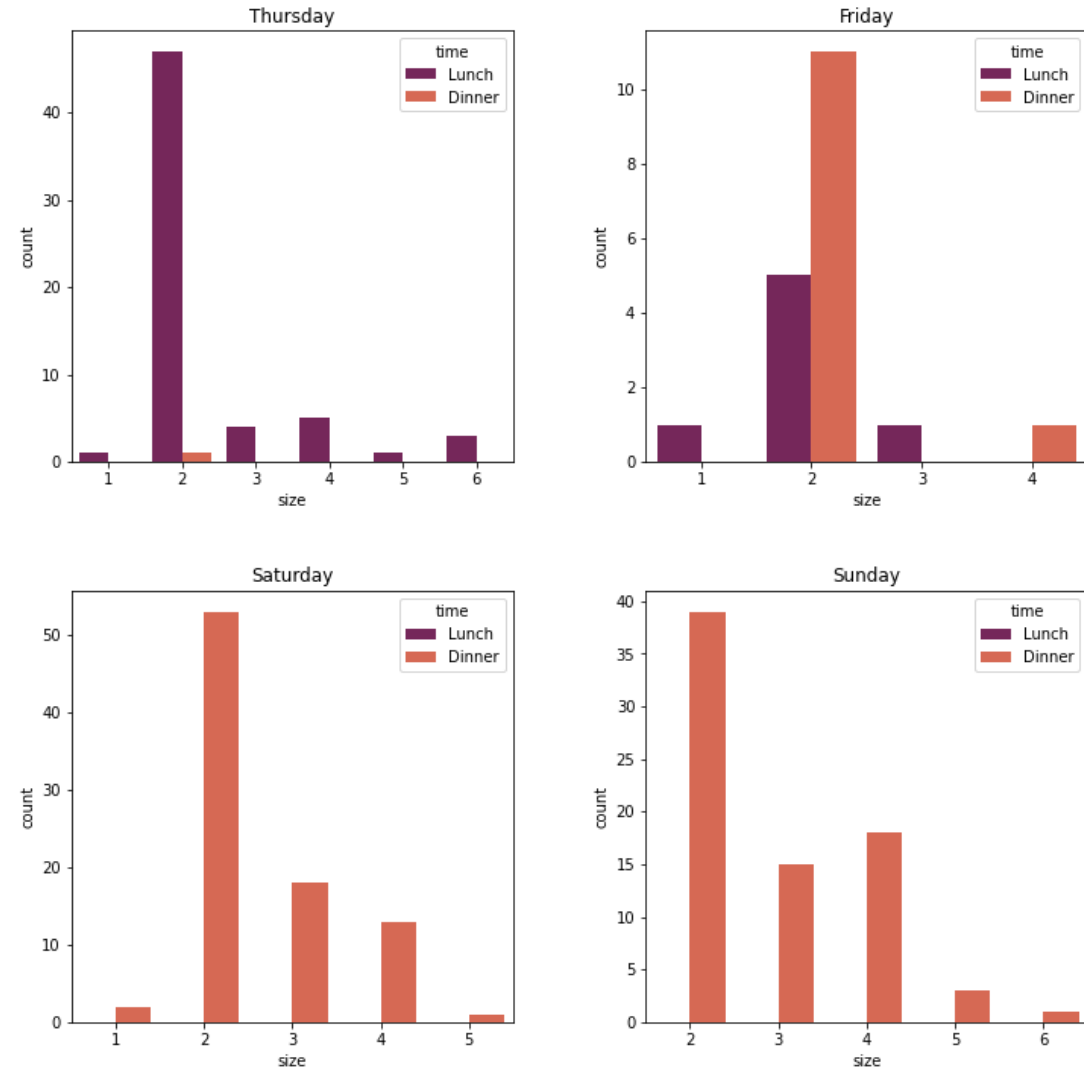
1. 손님의 방문 횟수를 countplot을 이용하여 요일별 식사시간대에 따라 각각 나타내라
2. 요일별 식사시간에 따른 팀 별 손님의 인원수를 countplot을 이용하여 나타내라

문제 2

day count with time



Size by time for each day



정답

1번 해답

```
sns.countplot(x = 'day', data = tips, palette='rocket', hue = 'time', saturation = 0.8)

plt.title("day count with time", fontsize = 20)
plt.show()
```

2번 해답

```
fig, ax = plt.subplots(2,2, figsize = (12,12))
plt.subplots_adjust(wspace = 0.3, hspace=0.3)
fig.suptitle('Size by time for each day')

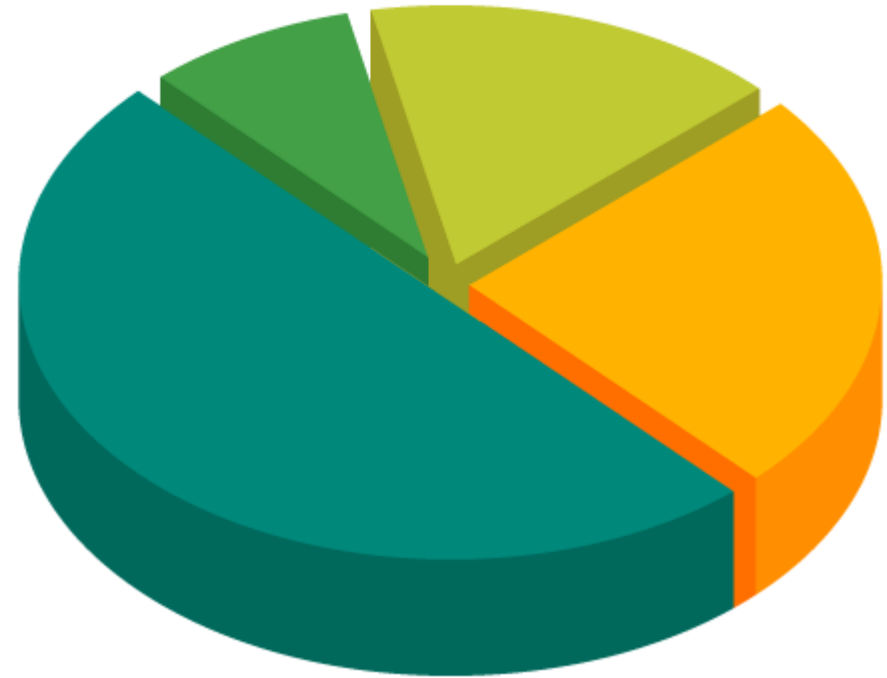
c1 = sns.countplot(x = 'size', data = tips[tips["day"] == "Thur"], palette='rocket', hue = 'time', saturation = 0.8, ax = ax[0,0])
c2 = sns.countplot(x = 'size', data = tips[tips["day"] == "Fri"], palette='rocket', hue = 'time', saturation = 0.8, ax = ax[0,1])
c3 = sns.countplot(x = 'size', data = tips[tips["day"] == "Sat"], palette='rocket', hue = 'time', saturation = 0.8, ax = ax[1,0])
c4 = sns.countplot(x = 'size', data = tips[tips["day"] == "Sun"], palette='rocket', hue = 'time', saturation = 0.8, ax = ax[1,1])

c1.set(title = "Thursday")
c2.set(title = "Friday")
c3.set(title = "Saturday")
c4.set(title = "Sunday")
plt.show()
```

Pie chart

범주별 데이터의 양을 파이모양으로 표시,
비율에 대한 정보를 한 눈에 보기 좋음.

Pie Chart



문제 3

Titanic 데이터를 이용하여 생존자의 비율을 pie chart로 나타내라.



문제 3

정답

```
titanic = sns.load_dataset("titanic")

survived = titanic['survived'].value_counts()
label = ['dead', "survived"]

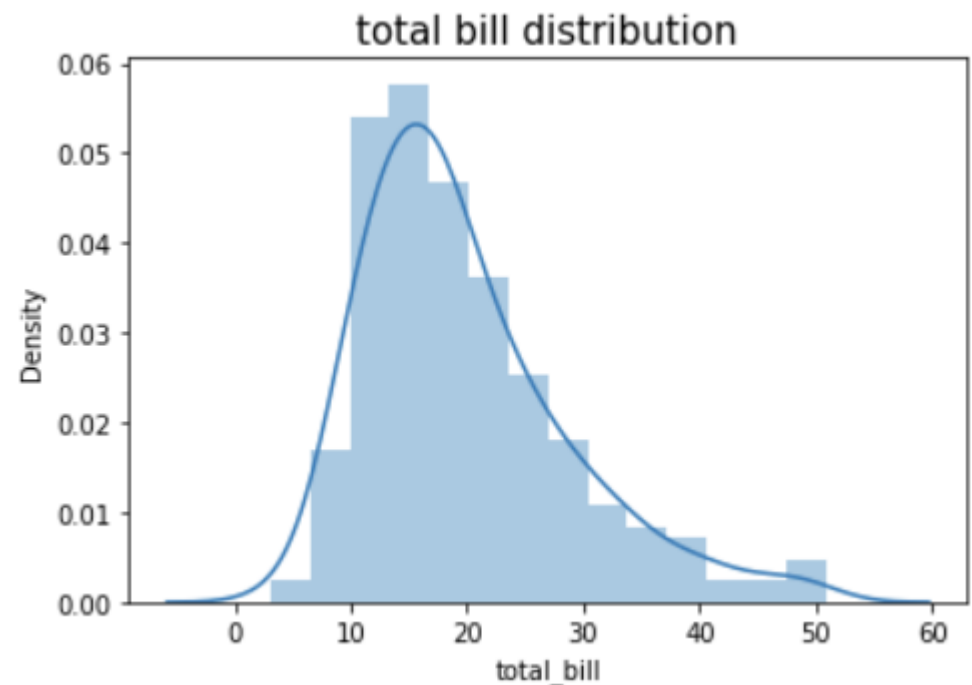
explode = [0.08, 0.03]
colors = ['#ae0400', '#43ae32']
wedgeprops={'width': 0.7, 'edgecolor': 'w', 'linewidth': 5}
plt.pie(survived, labels=label,
        autopct='%1.2f%%', startangle = 30,
        explode = explode, colors = colors, wedgeprops = wedgeprops,
        shadow = True)

plt.title("survived pie", fontsize = 18)
plt.show()
```

histogram

도수분포표의 시각화 기법.

계급구간(cm)	도수
161.5 이상 165.5 미만	6
165.5 이상 169.5 미만	12
169.5 이상 173.5 미만	18
173.5 이상 177.5 미만	11
177.5 이상 181.5 미만	8
합계	55

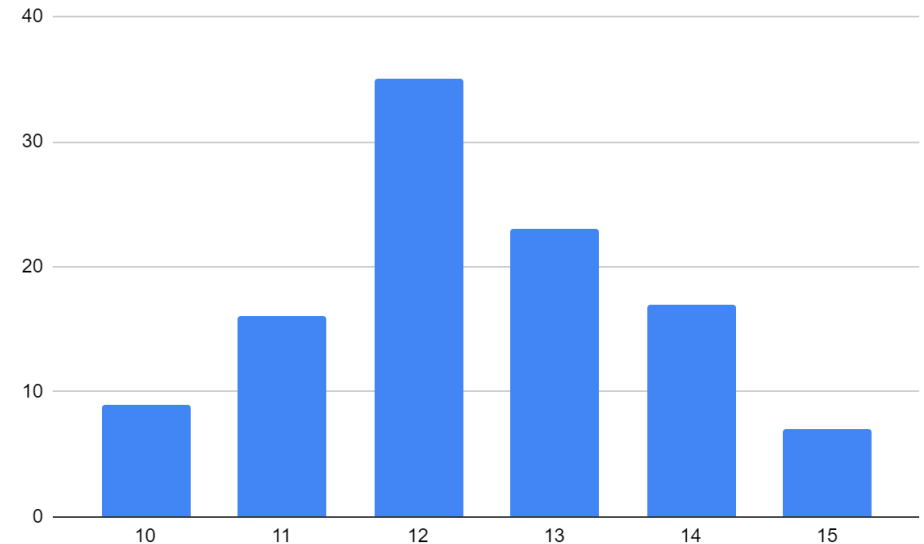


밀도추정 / kde

주어진 데이터로 '분포' 를 알고 싶다.....왜?

분포가 주어지면 입력값에 대한 결과값을 '통계적' 으로 추정 가능하기 때문

시간	방문 손님 수
10	9
11	16
12	35
13	23
14	19
15	7



밀도추정 / Kde

모수추정(Parametric)

- 정해진 틀에 끼워 맞추기
- Ex) 학생들의 키가 정규분포를 따른다 가정
- 데이터 모양이 아주 단조롭고 뻔할때 사용
- Ex) 정규분포, 감마분포, 와이블분포...

비모수추정(non-Parametric)

- 데이터로 부터 분포를 제작하기
- Ex) 데이터가 우리가 아는 어떤 분포도 따르지 않아...
- 지 멋대로 생겨 먹은 데이터에 사용
- Ex) 히스토그램, Kde, KNN

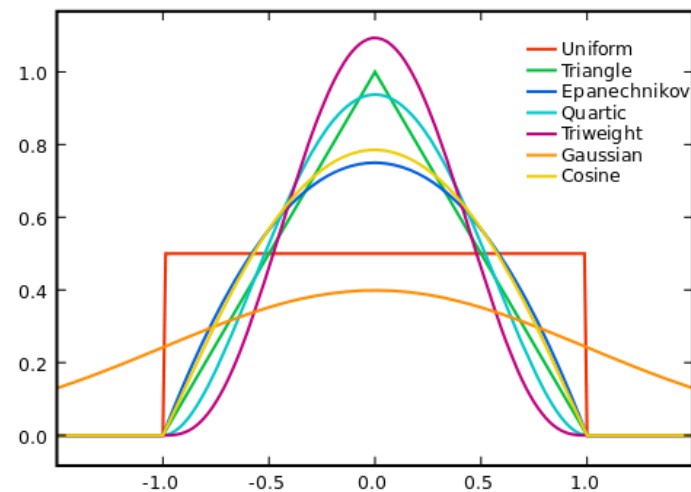
밀도추정 / Kde

Kernel function

$$\int_{-\infty}^{+\infty} K(u) du = 1;$$

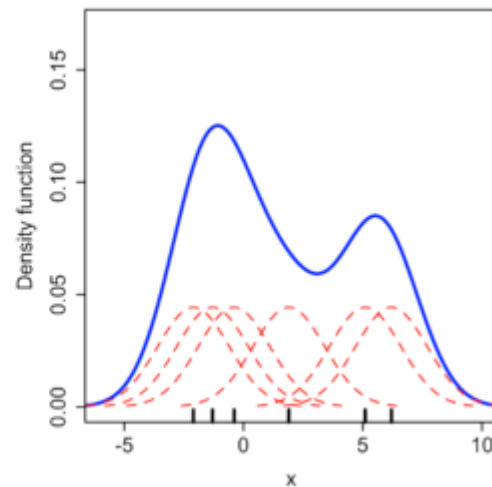
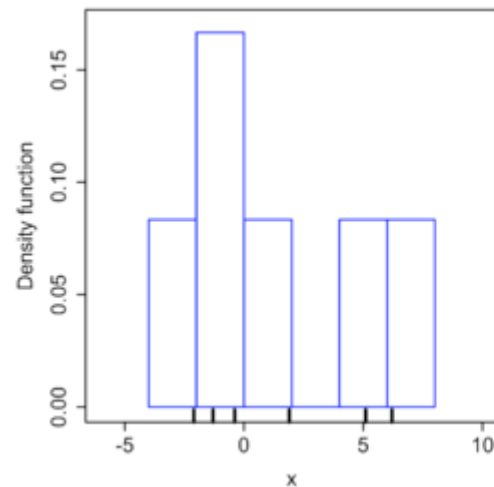
- Symmetry:

$$K(-u) = K(u) \text{ for all values of } u.$$



Pdf of Kernel function

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$



문제 4

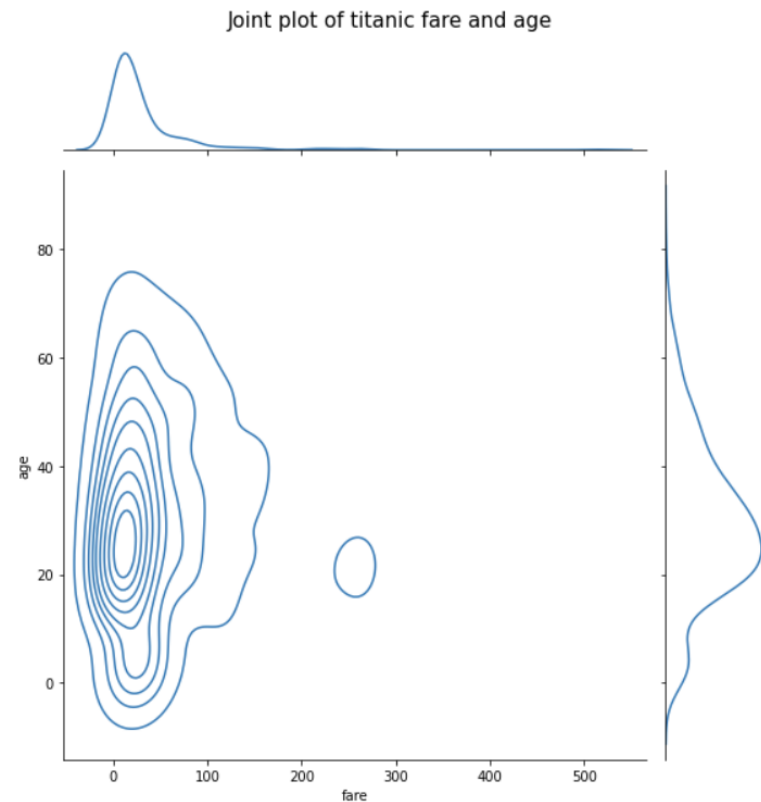
Titanic 데이터를 이용하여 운임요금(fare)에 따른 연령(age)을 kde 방법으로 jointplot 하고 분석하라.

문제 4

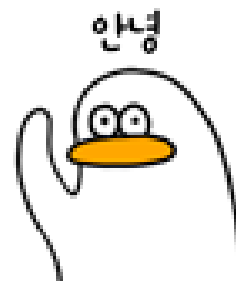
정답

```
joint = sns.jointplot(data = titanic, x = "fare", y="age", kind = 'kde', height = 8)

joint.set_axis_labels(xlabel = "fare", ylabel = "age")
joint.fig.suptitle("Joint plot of titanic fare and age", fontsize = 15, y = 1.03)
plt.show()
```



끼
트



담에뵈시당