데이터 시각화 (2024)

데이터과학부 정진명
(jmjung@suwon.ac.kr, 글로벌경상관 918호)

5 주차

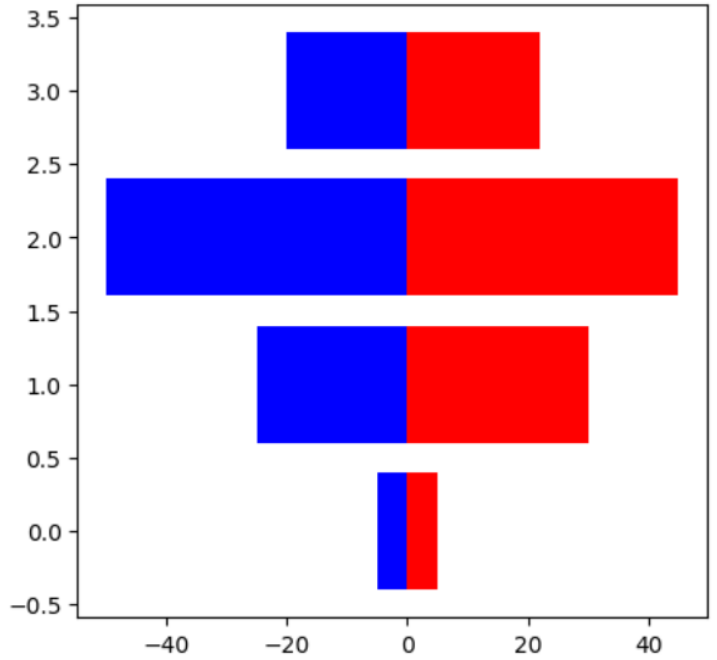
Contents

- bar (Cont'd)
- pie
- histogram
- boxplot

bar

양방향 막대차트 그리기

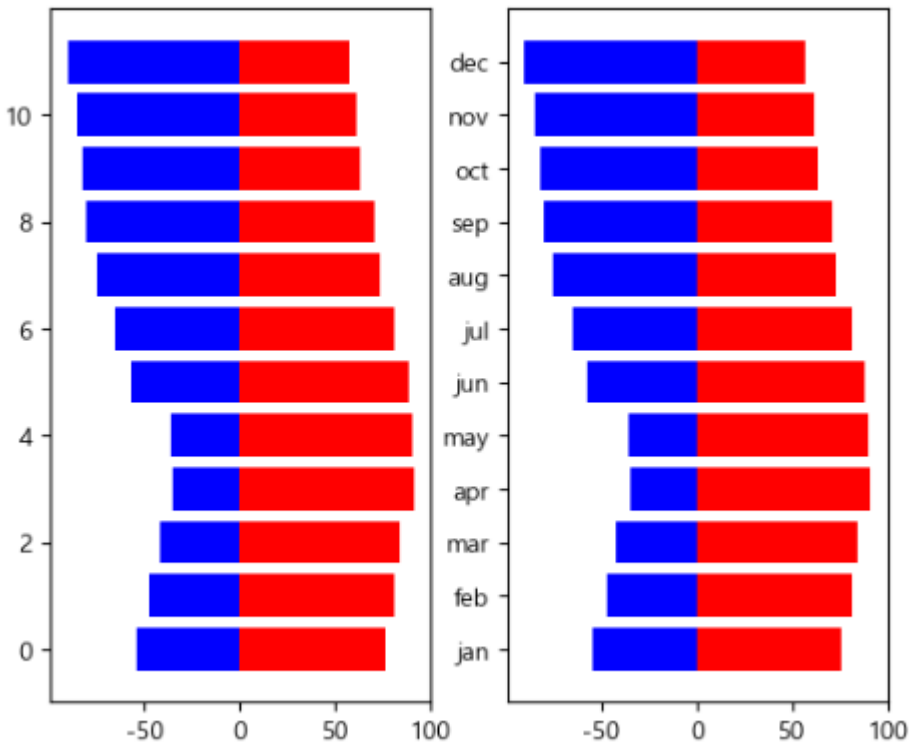
```
fig=plt.figure(figsize=(5,5), dpi=100)
ax=fig.subplots()
                          대전, 대구, 서울, 부산
women_pop=np.array([5,30,45,22])
men_pop=np.array([5,25,50,20])
X=np.arange(4)

ax.barh(X,women_pop, color='r')
ax.barh(X,-men_pop, color='b')
```



양방향 막대차트 그리기

- dat_bar.txt를 읽어서 아래와 같이 양방향 막대를 그리시오

```
fig=plt.figure(figsize=(6,5), dpi=100)
ax1, ax2=fig.subplots(1,2)

data=pd.read_table('data/dat_bar.txt',sep='\t', index_col=0)
data

## ax1
X=np.arange(len(data))
ax1.barh(X,data['iphone'], color='r')
ax1.barh(X,-data['galaxy'], color='b')

## ax2
ax2.barh(data.index,data['iphone'], color='r')
ax2.barh(data.index,-data['galaxy'], color='b')
```



pie

원형 차트 (pie)

matplotlib.pyplot.pie(*x*, *explode=None*, *labels=None*, *colors=None*, *autopct=None*, *pctdistance=0.6*, *shadow=False*, *labeldistance=1.1*, *startangle=None*, *radius=None*, *counterclock=True*, *wedgeprops=None*, *textprops=None*, *center=(0, 0)*, *frame=False*, *rotatelabels=False*, ***, *data=None*) ¶ [source]

Plot a pie chart.

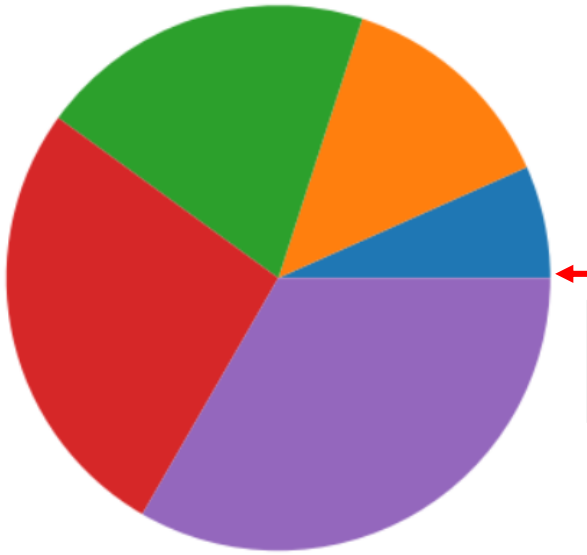
Make a pie chart of array *x*. The fractional area of each wedge is given by `x/sum(x)`. If `sum(x) < 1`, then the values of *x* give the fractional area directly and the array will not be normalized. The resulting pie will have an empty wedge of size `1 - sum(x)`.

The wedges are plotted counterclockwise, by default starting from the x-axis.

- pie chart
  - 수량의 상대적인 중요성을 비교할 때 주로 사용

원형 차트 (pie)

```
fig=plt.figure(figsize=(5,5), dpi=100)
ax=fig.subplots()

data=np.arange(1,6)

_=ax.pie(data)
```

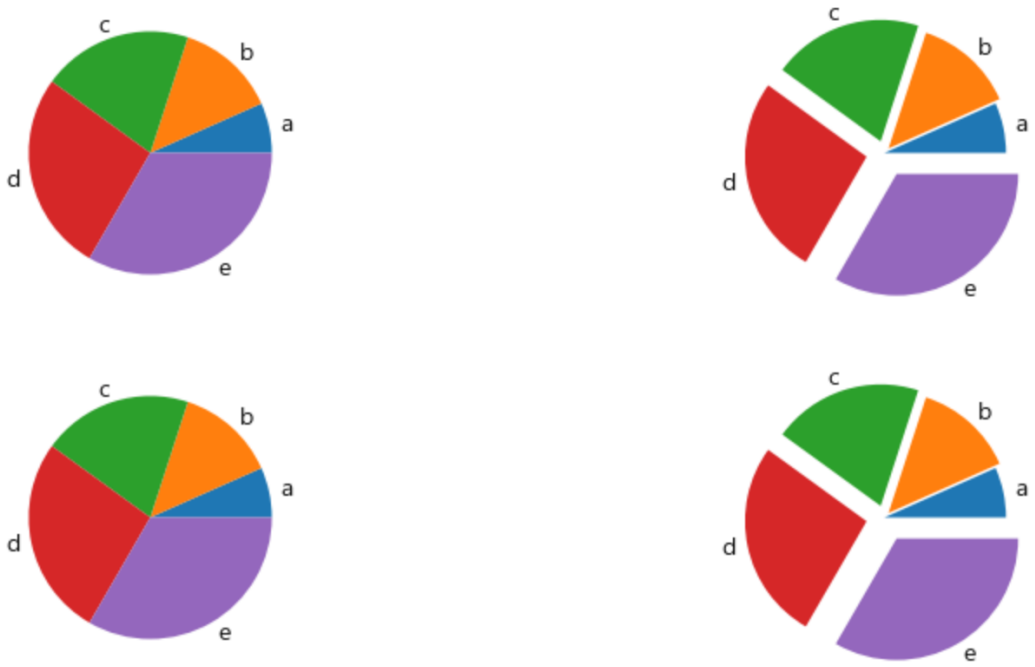


첫번째 data가
plot되는 시작점

값들이 전체 합이 1 이상인 경우: -
→ 합이 1이 되도록 normalization
을 수행 후 pie plot을 그림

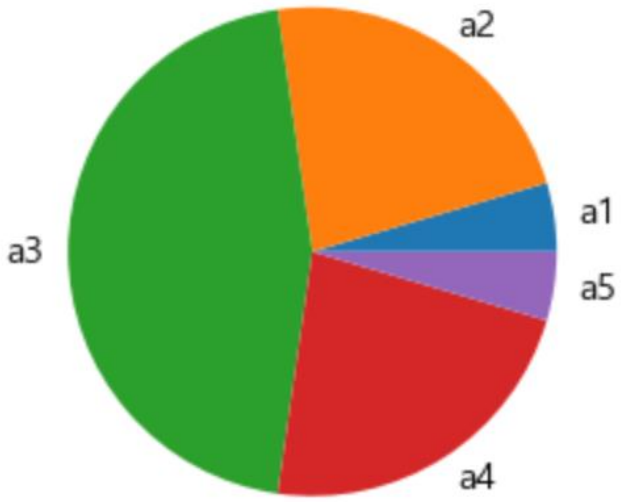
원형 차트 (pie) with labels, explode

```
fig=plt.figure(figsize=(10,5), dpi=100)
ax1, ax2, ax3, ax4=fig.subplots(2,2).flatten()

data=pd.Series(data=np.arange(1,6), index=list('abcde'))

_=ax1.pie(x=data.values, labels=data.index)
_=ax2.pie(x=data.values, labels=data.index, explode=np.array([0,1,2,3,4])*0.05)
_=data.plot.pie(ax=ax3)
_=data.plot.pie(ax=ax4, explode=np.array([0,1,2,3,4])*0.05)
```



실습 1

- Data를 직접 입력하여, 아래와 같은 pie 그래프를 그리시오



1. 색깔은 제공되는 색깔 사용
2. 비율 ➔ 파랑:주황:초록:빨강:보라=1:5:10:5:1

histogram

히스토그램 (hist)

```
matplotlib.pyplot.hist(x, bins=None, range=None, density=False, weights=None, cumulative=False, bottom=None, histtype='bar',
align='mid', orientation='vertical', rwidth=None, log=False, color=None, label=None, stacked=False, *, data=None, **kwargs)
```
[source]

Plot a histogram.

Compute and draw the histogram of *x*. The return value is a tuple (*n*, *bins*, *patches*) or ([*n0*, *n1*, ...], *bins*, [*patches0*, *patches1*, ...]) if the input contains multiple data. See the documentation of the *weights* parameter to draw a histogram of already-binned data.

Multiple data can be provided via *x* as a list of datasets of potentially different length ([*x0*, *x1*, ...]), or as a 2-D ndarray in which each column is a dataset. Note that the ndarray form is transposed relative to the list form.

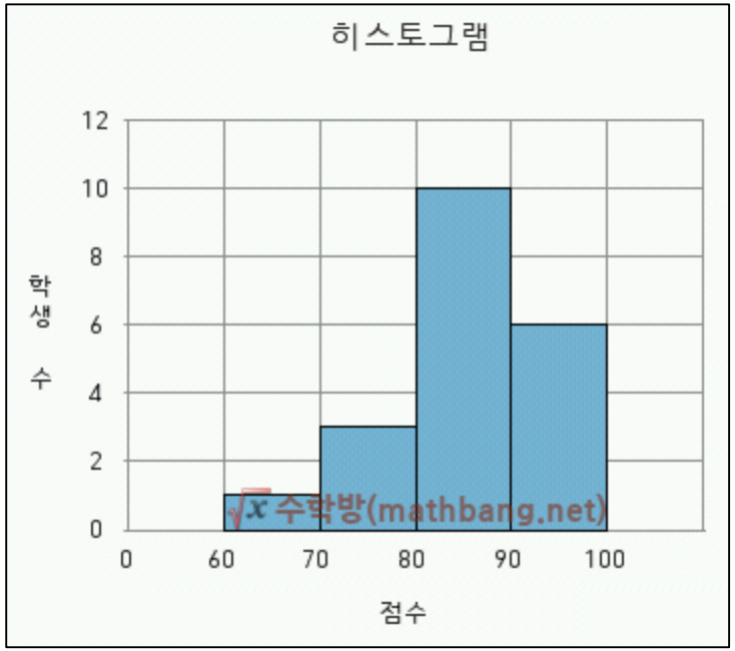
Masked arrays are not supported.

The *bins*, *range*, *weights*, and *density* parameters behave as in `numpy.histogram`.

- histogram:
  도수분포표를 그래프로 나타낸 것, 분포 확인 시 주로 사용

도수분포표

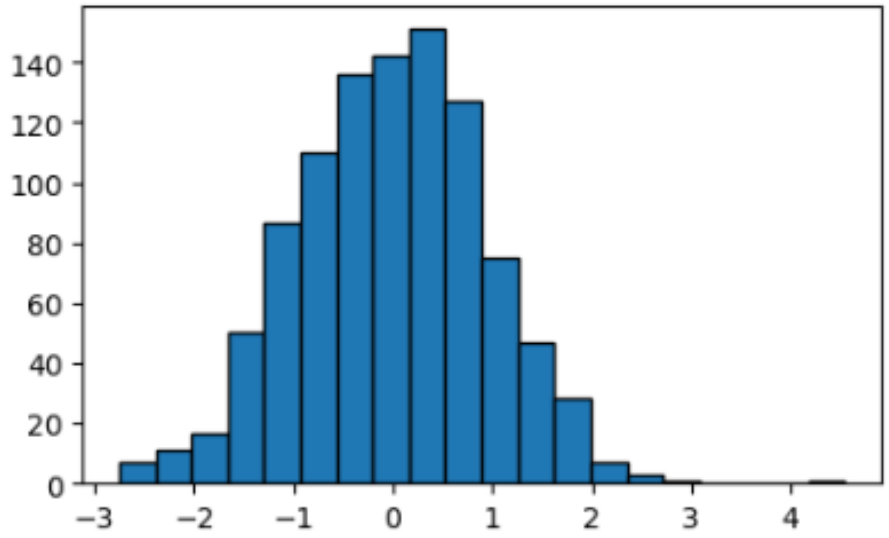
점수(점)	학생 수(명)
60 이상 ~ 70 미만	1
70 ~ 80	3
80 ~ 90	10
90 ~ 100	6
합계	20



히스토그램 (hist)

```
fig=plt.figure(figsize=(5,3), dpi=100)
ax=fig.subplots()

data=np.random.normal(size=1000)
data[:20]
data.min()
data.max()


_=ax.hist(data, bins=20, edgecolor='k')
```

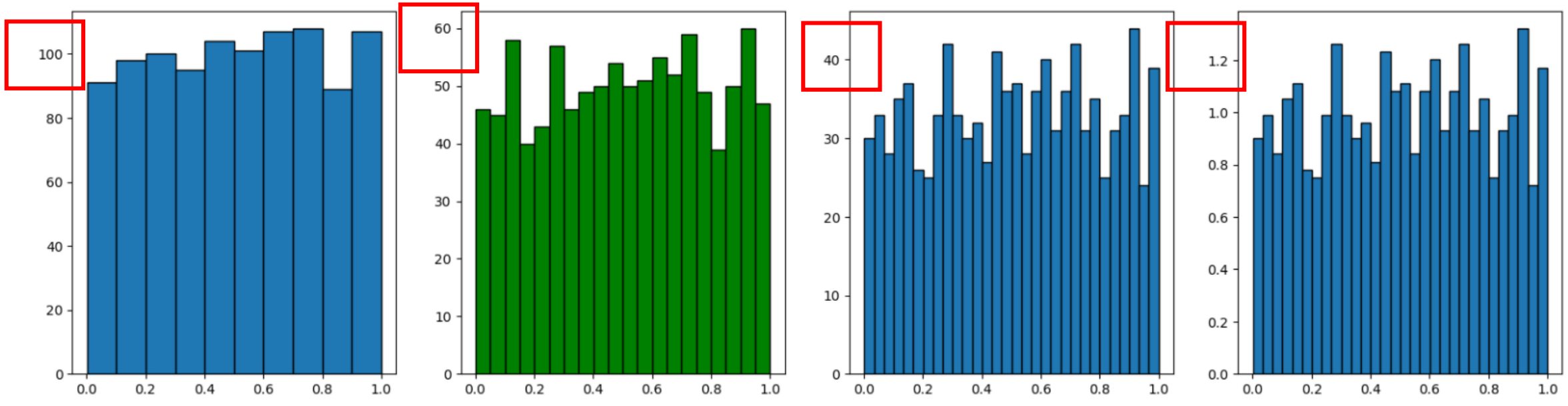
```
array([-0.93605566,  1.50566486,  0.55440143, -1.59734149,  1.0468464 ,
        0.16742531, -0.64851687,  0.30993156, -0.57206381,  0.17550248,
       -0.57239769,  0.93203063, -0.66076909, -0.63328692,  0.20492814,
       -0.32005653, -0.1005496 , -1.74862407, -1.03429937, -0.07914048])


-2.7559272582482417


4.542060578896256
```



주어진 data의 최소값, 최대값 사이를 bin수 만큼 균등하게 나눈다.

히스토그램 (hist)

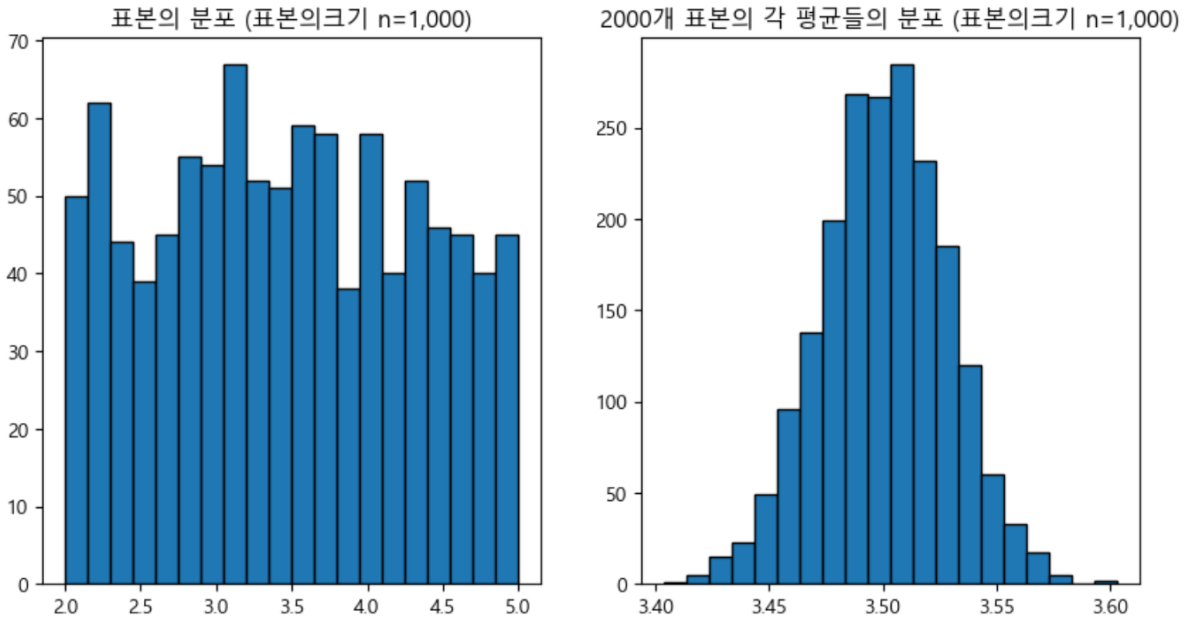
```
fig=plt.figure(figsize=(20,5), dpi=100)
axs=fig.subplots(1,4)

data=np.random.uniform(size=1000)

_=axs[0].hist(data, bins=10, edgecolor='k')
_=axs[1].hist(data, bins=20, edgecolor='k', color='g')
_=axs[2].hist(data, bins=30, edgecolor='k')
_=axs[3].hist(data, bins=30, edgecolor='k', density=True)
```



중심극한정리 예제

```
fig=plt.figure(figsize=(10,5), dpi=100)
axs=fig.subplots(1,2)

data=np.random.uniform(2,5,size=1000)

_=axs[0].hist(data, bins=20, edgecolor='k')
_=axs[0].set_title('표본의 분포 (표본의크기 n=1,000)')

data=np.random.uniform(2,5,size=(2000,1000))
m1=data.mean(axis=1)
m1.shape

_=axs[1].hist(m1, bins=20, edgecolor='k')
_=axs[1].set_title('2000개 표본의 각 평균들의 분포 (표본의크기 n=1,000)')
```

모집단의 분포 (모분포): uniform (2,5)



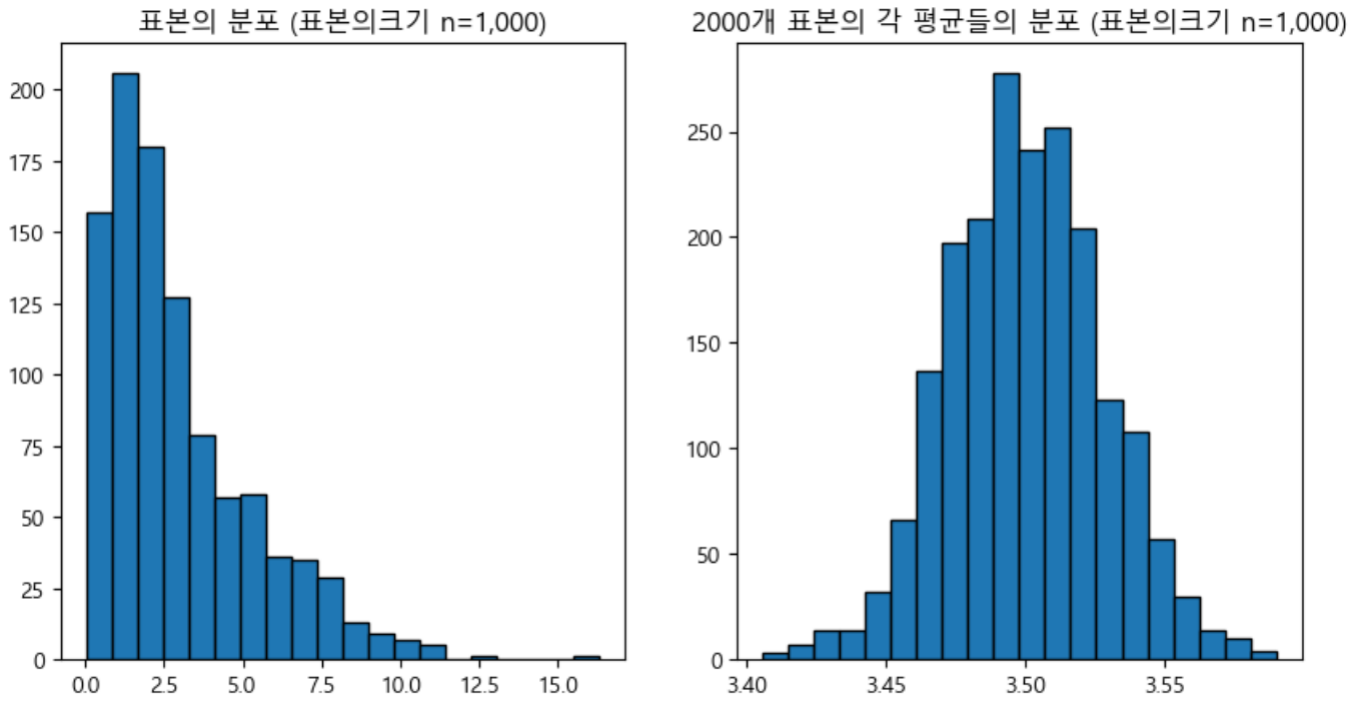
중심 극한 정리

중심극한정리

모집단의 분포가 무엇이든 관계없이 n이 크면 표본평균 $\overline{X}$의 분포는 근사적으로 정규분포가 된다. 즉, 평균 μ, 표준편차 σ인 임의의 모집단으로부터의 표본평균 $\overline{X}$는 n이 크면 근사적으로 평균 μ, 표준편차 $\sigma/\sqrt{n}$인 정규분포를 따른다. 따라서, $Z=\frac{\overline{X}-\mu}{\sigma/\sqrt{n}}$는 근사적으로 표준정규분포 $(N(0,\ 1))$를 따른다.

실습 2

모집단의 분포 (모분포): chisquare (df=3)

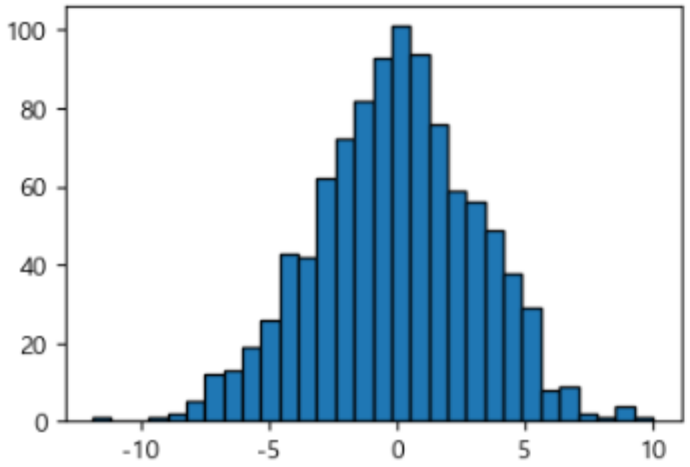


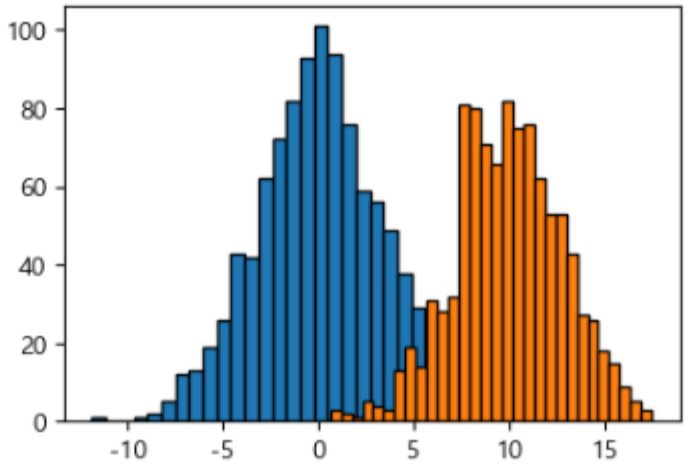
numpy.mean(), for loop 각각 사용해보기

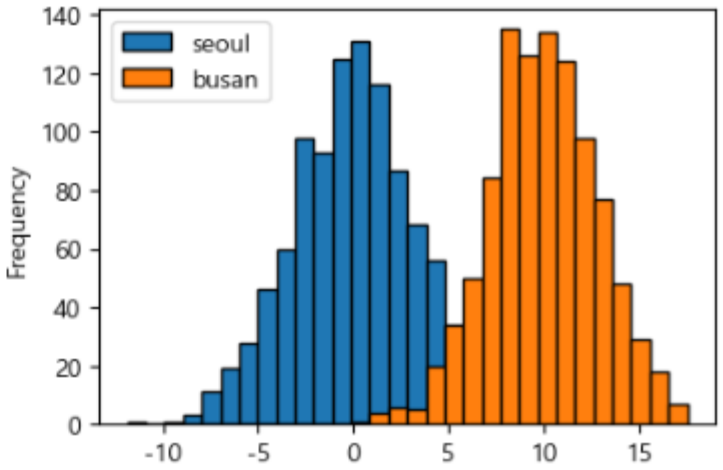
파일에서 읽어서 히스토그램 그리기

1) csv파일 'dat_hist.txt ' 을 읽어서,
seoul 데이터를 ax1에,
seoul, busan 데이터를 같이 ax2에,
pandas.plot.hist 함수를 사용하여 ax3에,
histogram으로 그리시오 (bin의 개수: 30)

```
fig=plt.figure(figsize=(15,3), dpi=100)
ax1,ax2,ax3=fig.subplots(1,3)

data=pd.read_table('data/dat_hist.txt',sep=',')
data

_=ax1.hist(data['seoul'], bins=30, edgecolor='k')
_=ax2.hist(data['seoul'], bins=30, edgecolor='k')
_=ax2.hist(data['busan'], bins=30, edgecolor='k')
_=data.plot.hist(ax=ax3, bins=30, edgecolor='k')
```

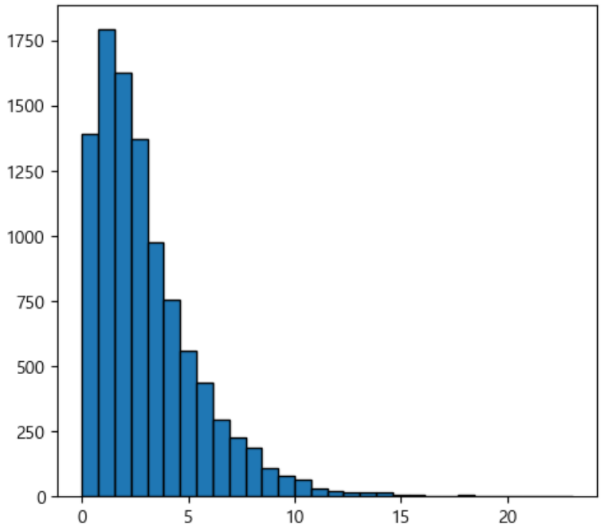


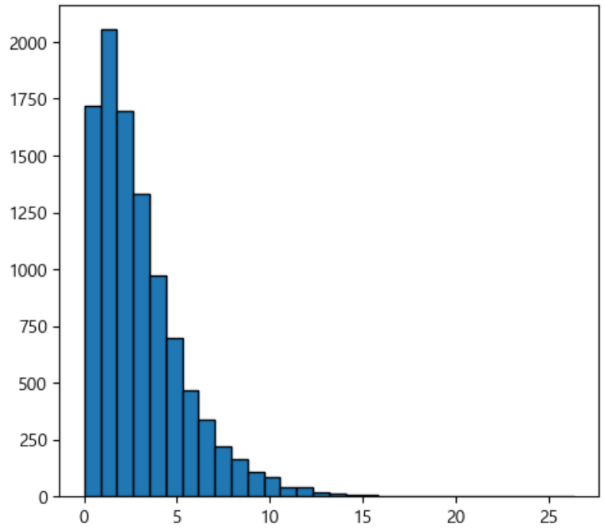




실습 3

- 평균 0, 표준편차 1의 정규분포를 따르는 확률 변수 X가 있을 때,
  Y=X**2+X**2+X**2의 분포를 히스토그램으로 그리시오 (cf: chisquare distribution)





연속확률변수 X의 확률밀도함수 $f(x)$가 다음과 같을 때,

$$f(x;v)=\frac{1}{2^{\frac{v}{2}}\Gamma\left(\frac{v}{2}\right)}x^{\frac{v}{2}-1}e^{-\frac{x}{2}},\ (x>0)$$

X는 자유도 v인 '카이제곱 분포'를 따른다고 한다.

확률변수 X가 카이제곱 분포를 따르면 다음과 같이 표현합니다.

확률변수 X가 카이제곱 분포를 따르면

$X \sim \chi^2(v)$ 로 표현한다.

정의 [편집]

양의 정수 k가 주어졌다고 하고, k개의 독립적이고 표준정규분포를 따르는 확률변수 $X_1, \cdots, X_k$를 정의하자.

그렇다면 **자유도 k의 카이제곱 분포**는 확률변수

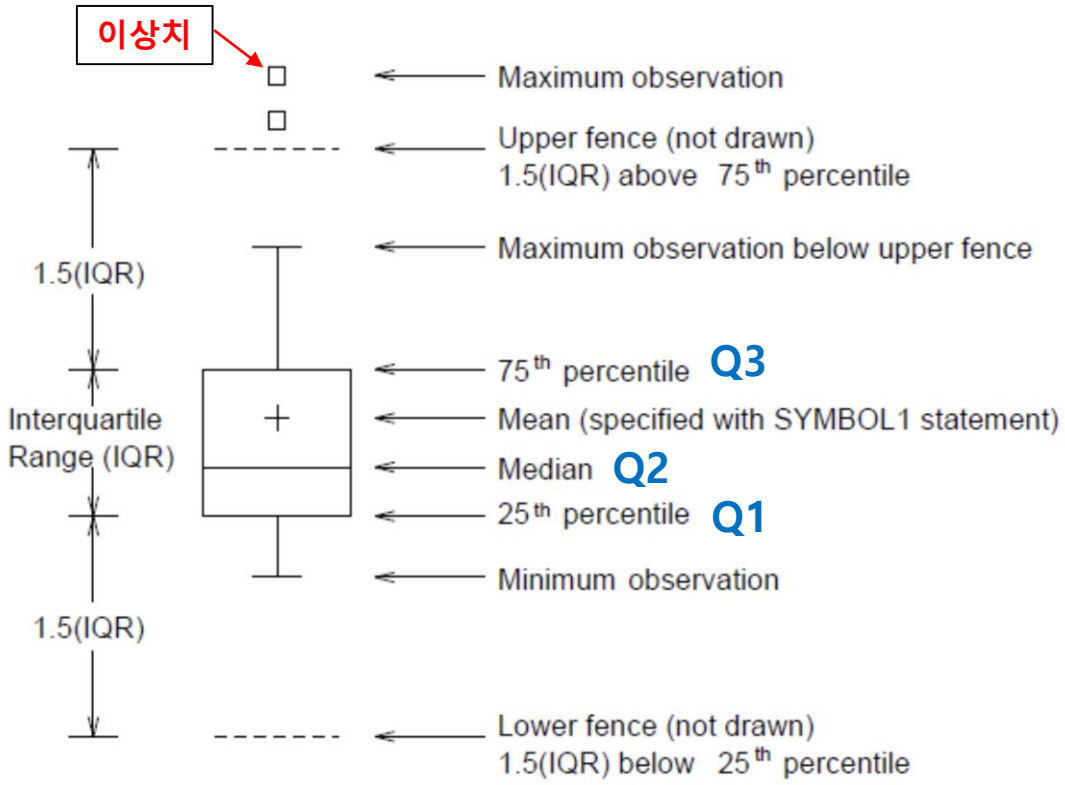
$$Q=\sum_{i=1}^{k}X_i^2$$

의 분포이다. 즉, $Q \sim \chi_k^2$ 이다.

boxplot

상자그림 그리기 (boxplot)

- boxplot(상자수염그림):
  - 최소값(min), 제1사분위(Q1) , 제2사분위(Q2), 제3사분위(Q3), 최대값(max) 표현
  - 수치형 자료 표현



- 제 2사분위는 중앙값(median)또는 중위수라고 하며, 가장 가운데 있는 값을 말한다. (평균이 아니다.)

상자그림 그리기 (boxplot)

matplotlib.pyplot.boxplot

matplotlib.pyplot.boxplot(*x*, *notch=None*, *sym=None*, *vert=None*, *whis=None*, *positions=None*, *widths=None*, *patch_artist=None*, *bootstrap=None*, *usermedians=None*, *conf_intervals=None*, *meanline=None*, *showmeans=None*, *showcaps=None*, *showbox=None*, *showfliers=None*, *boxprops=None*, *labels=None*, *flierprops=None*, *medianprops=None*, *meanprops=None*, *capprops=None*, *whiskerprops=None*, *manage_ticks=True*, *autorange=False*, *zorder=None*, ***, *data=None*) ¶ [source]

Make a box and whisker plot.

Make a box and whisker plot for each column of *x* or each vector in sequence *x*. The box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers extend from the box to show the range of the data. Flier points are those past the end of the whiskers.

Parameters:

x : Array or a sequence of vectors.

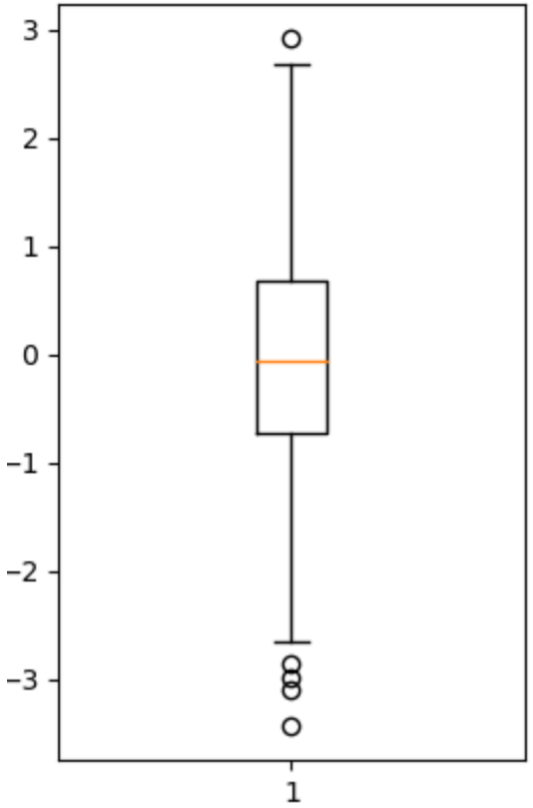
The input data.

notch : bool, default: False

상자그림 그리기 (boxplot)

```
fig=plt.figure(figsize=(3,5), dpi=100)
ax=fig.subplots()

data1=np.random.normal(size=1000)
ax.boxplot(data1)
```

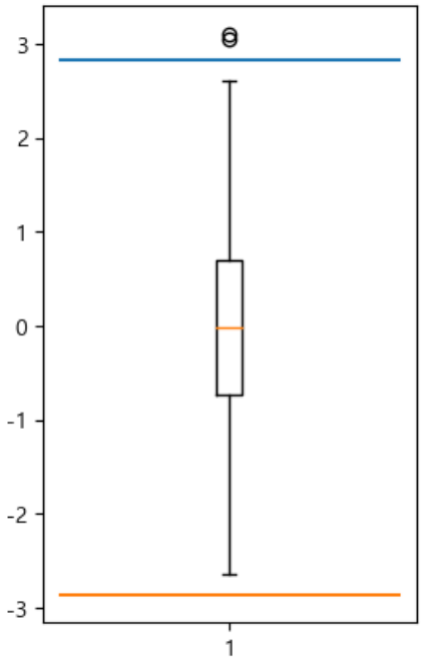
표준정규분포에서 값 1000개 sampling

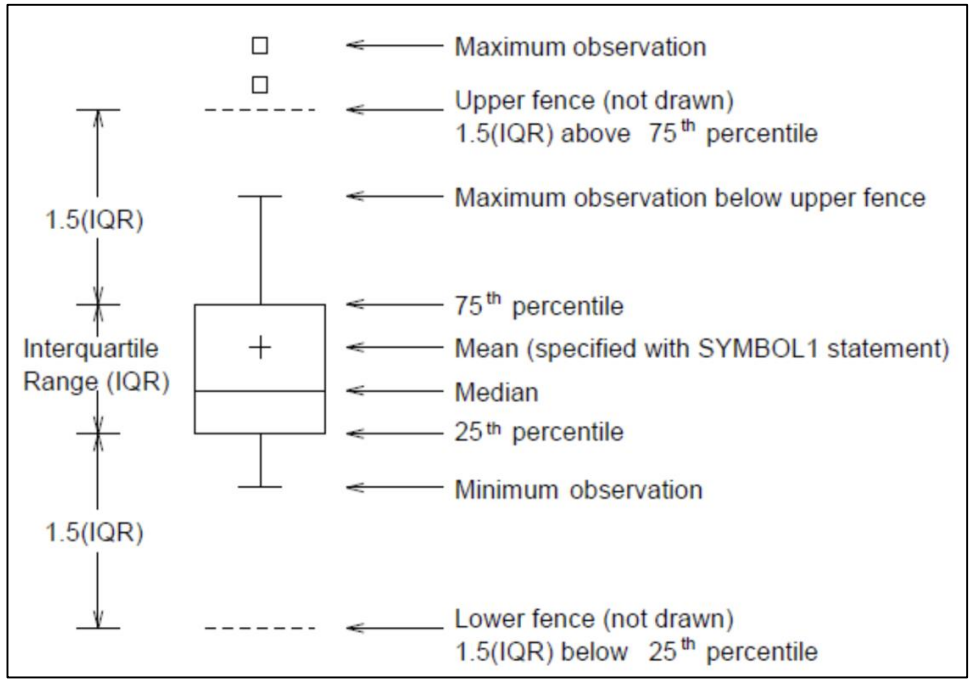


상자그림 그리기 (boxplot)

- boxplot: upper, lower fence 표시하기

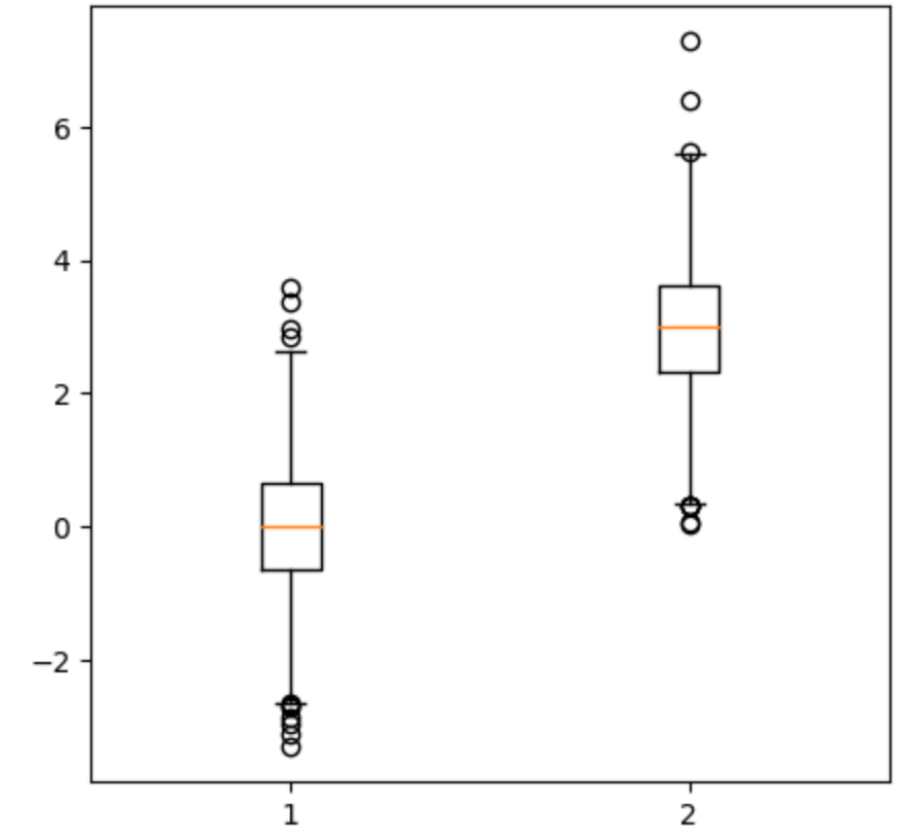
```
fig=plt.figure(figsize=(3,5), dpi=100)
ax=fig.subplots()

data1=np.random.normal(size=1000)
_=ax.boxplot(data1)

q75=np.quantile(data1,0.75)
q25=np.quantile(data1,0.25)

iqr=q75-q25
up_fence=q75+1.5*iqr
lw_fence=q25-1.5*iqr

ax.plot([-0,2],[up_fence,up_fence])
ax.plot([-0,2],[lw_fence,lw_fence])
```



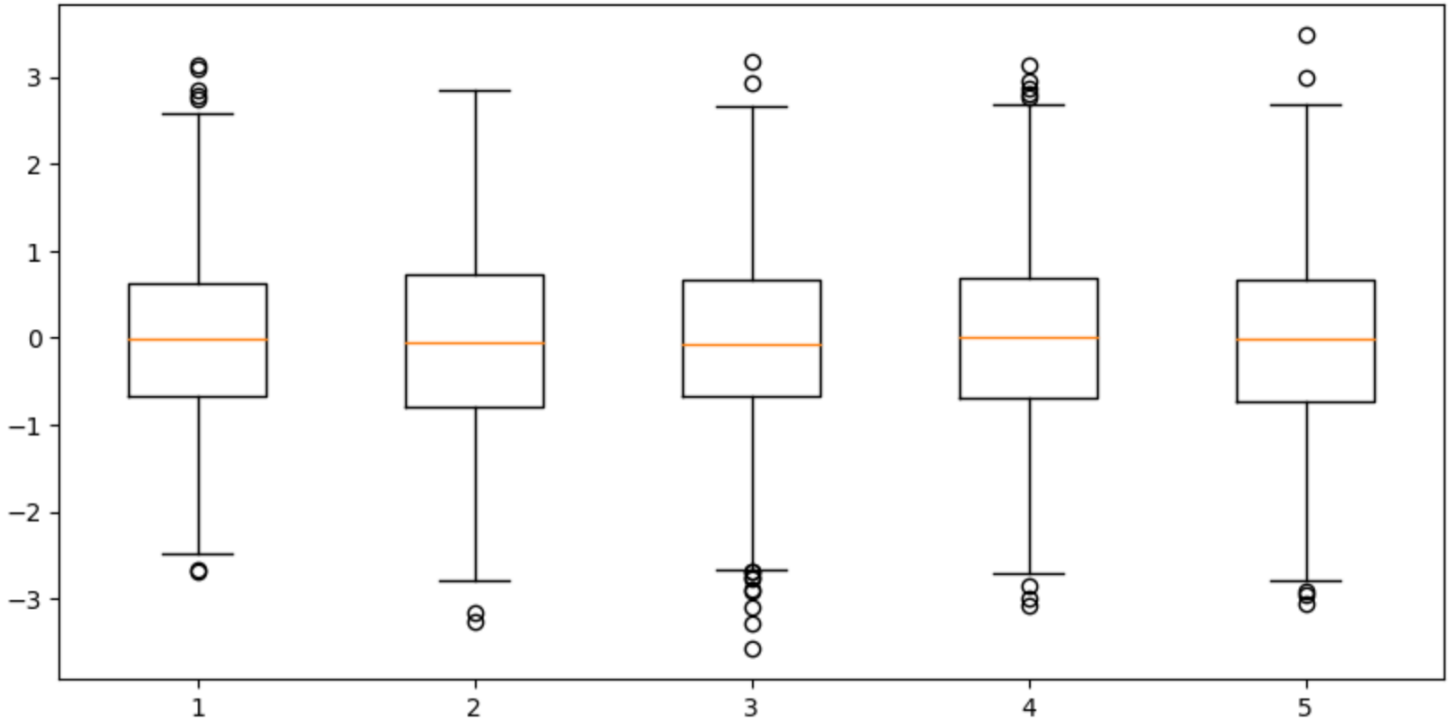


상자그림 그리기 (boxplot)

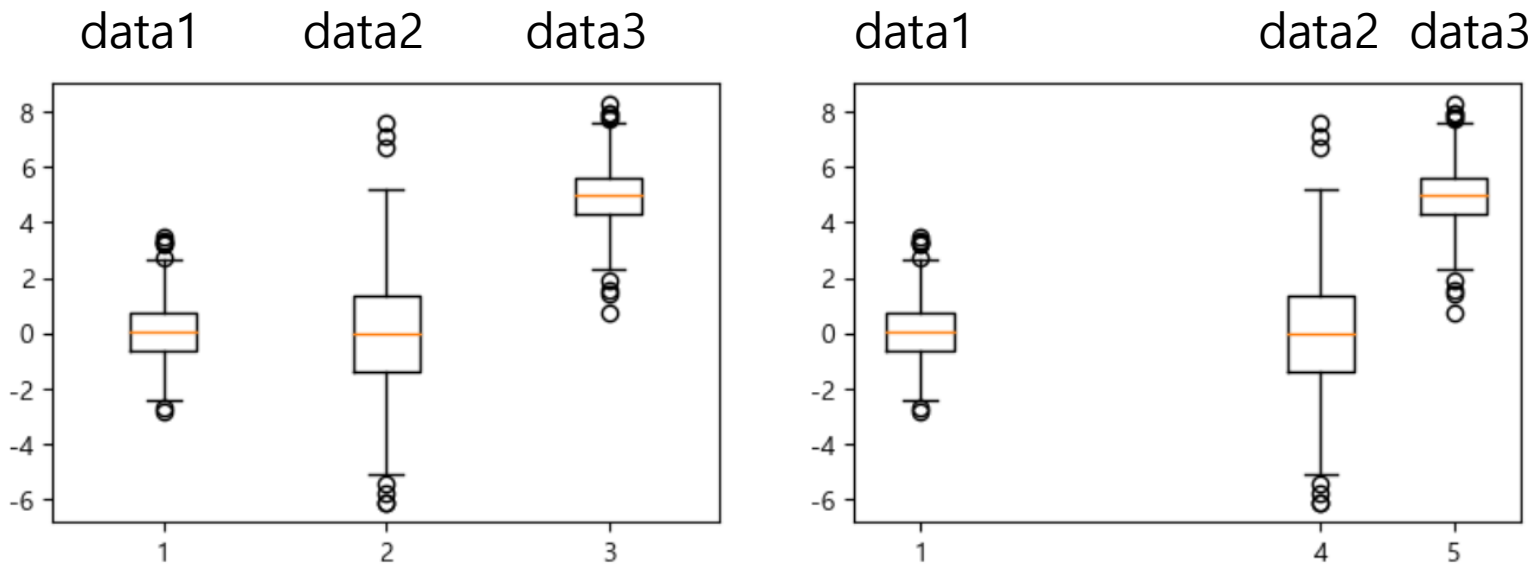
```
fig=plt.figure(figsize=(5,5), dpi=100)
ax=fig.subplots()

data1=np.random.normal(size=1000)
data2=np.random.normal(loc=3,size=1000)
_=ax.boxplot([data1, data2])
```




```
fig=plt.figure(figsize=(10,5), dpi=100)
ax=fig.subplots()

data1=np.random.normal(size=(1000,5))
_=ax.boxplot(data1)
```



실습 4

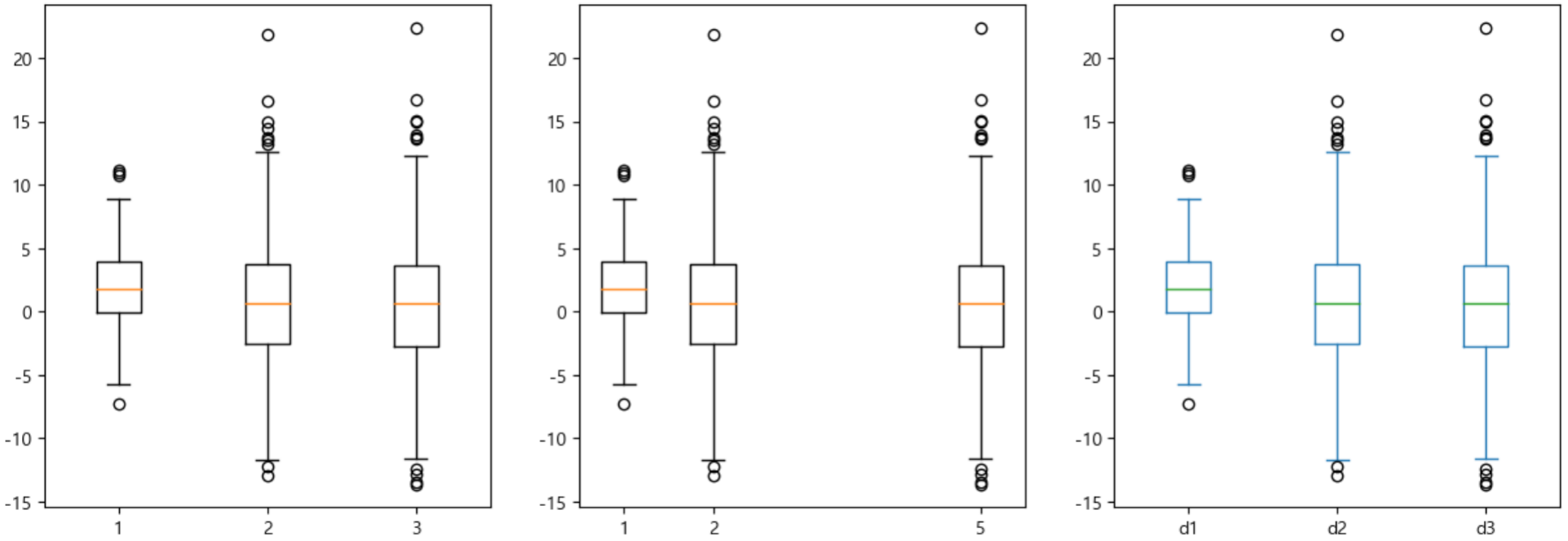


위 그림과 같이 3개의 data에 각각의 boxplot을 그리시오 (ax2: boxplot parameter "positions" 사용)

- data 1) 평균이 0, 표준편차 1의 정규분포에서 샘플링한 1000개의 값
- data 2) 평균이 0, 표준편차 2의 정규분포에서 샘플링한 1000개의 값
- data 3) 평균이 5, 표준편차 1의 정규분포에서 샘플링한 1000개의 값

파일로부터의 데이터 boxplot 그리기

```
fig=plt.figure(figsize=(15,5), dpi=100)
axs=fig.subplots(1,3)

data1=pd.read_table('data/dat_hist2.txt', sep='\t')
data1

_=axs[0].boxplot([data1['d1'],data1['d2'],data1['d3']])
_=axs[1].boxplot(data1, positions=[1,2,5])
_=data1.plot.box(ax=axs[2])
```



Q & A

Thank you