01wk-1

9/10/22

,

https://youtube.com/playlist?list=PLQqh36zP38-yhKDR2mVQyRQmesR0UIJBF

import

```
import matplotlib.pyplot as plt
import numpy as np
```

boxplot

motivating example

(1) : ?

- A B . A 79.1 B 78.3 .

```
y1=[75,75,76,76,77,77,79,79,79,98] # A
y2=[76,76,77,77,78,78,80,80,80,81] # B
```

```
np.mean(y1),np.mean(y2)
```

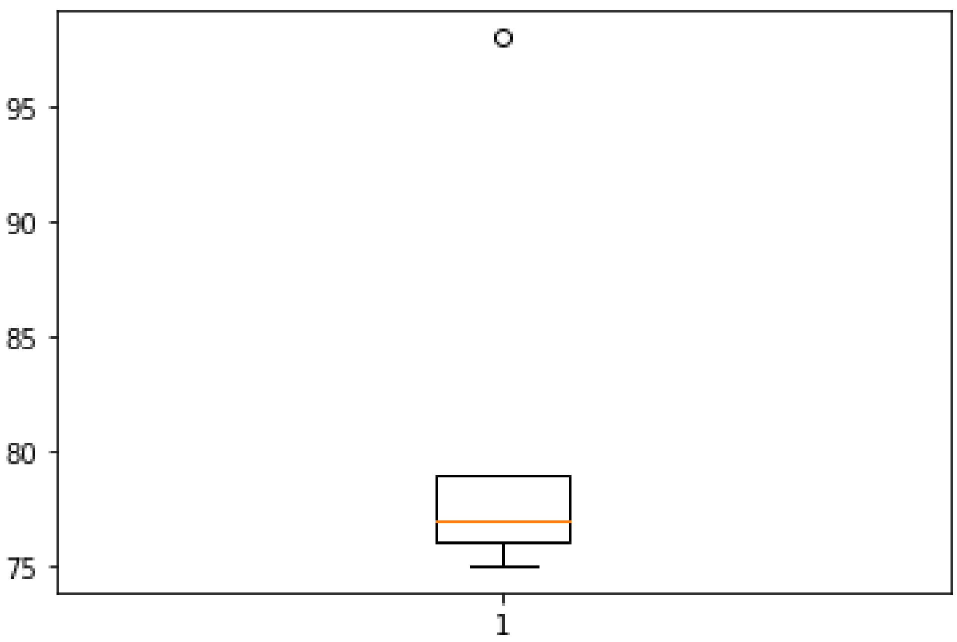
```
(79.1, 78.3)
```

- : A .
- A (=A) . 98 A A B .
- : . .

matplotlib boxplot

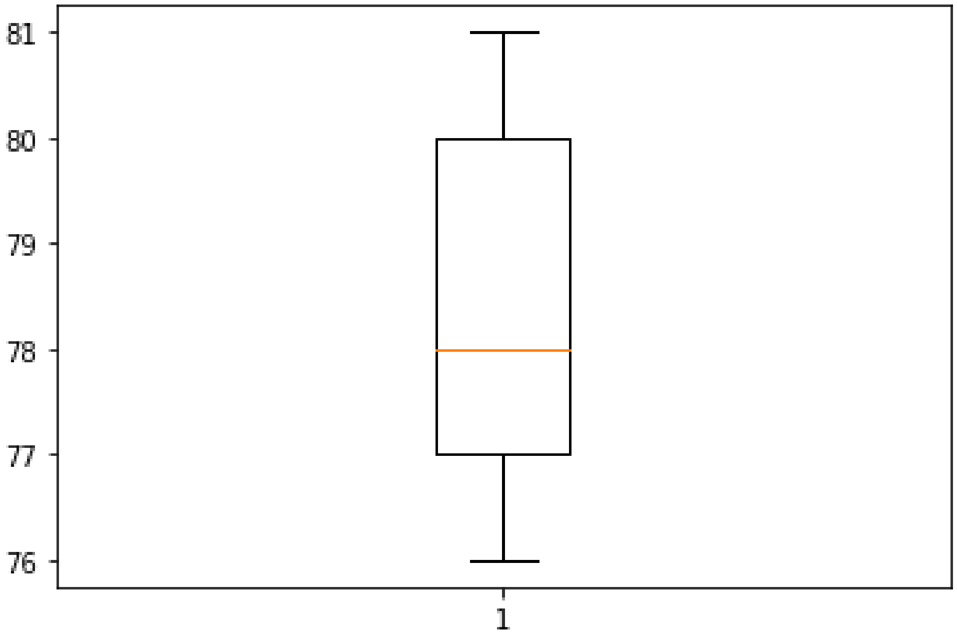
- A

```
plt.boxplot(y1);
```



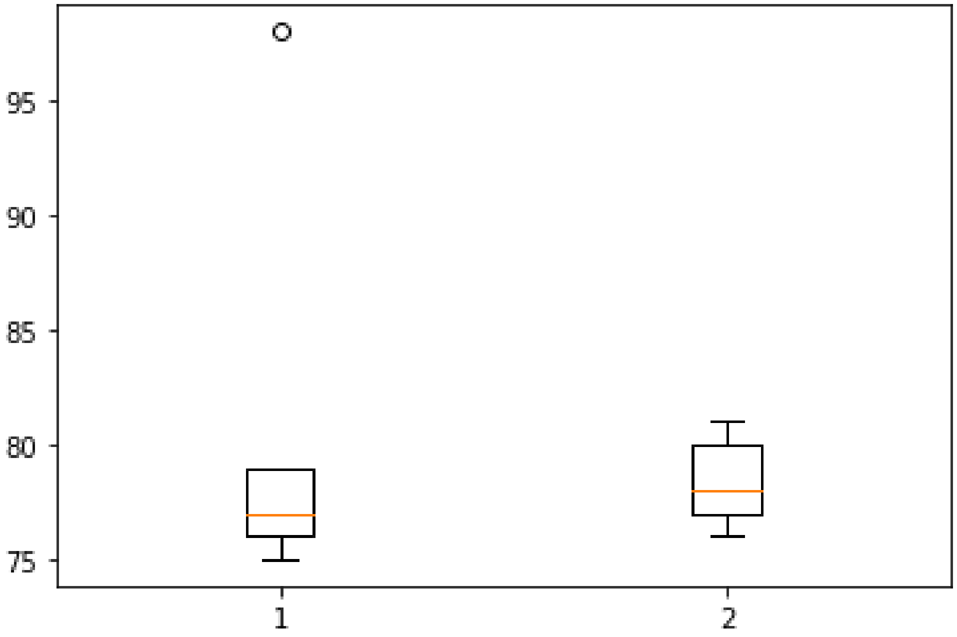
- B

```
plt.boxplot(y2);
```



- A B .

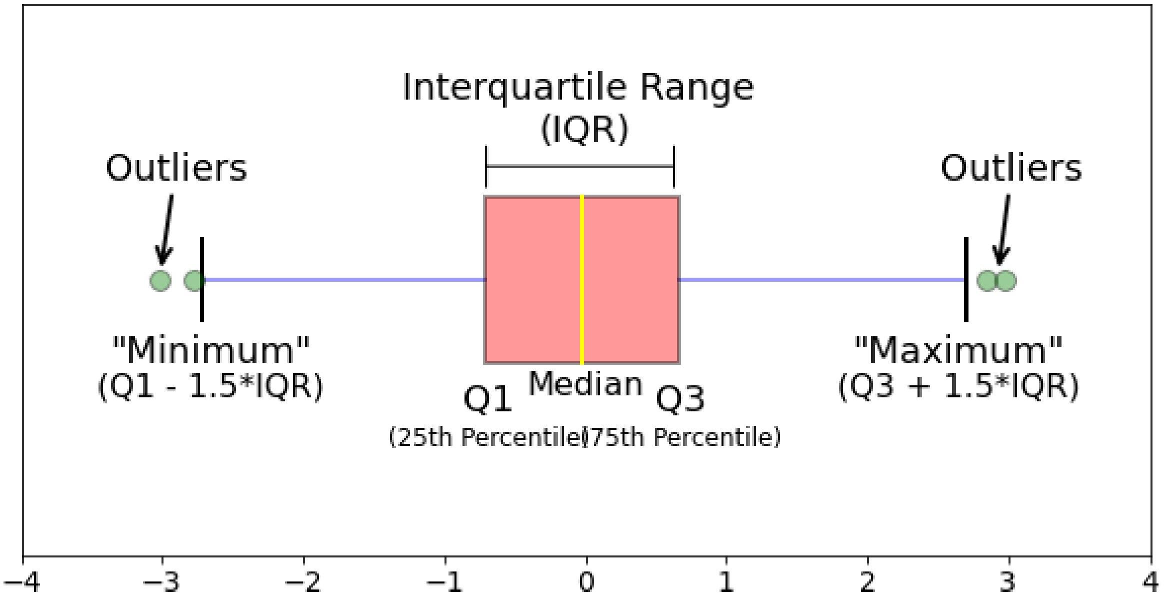
```
plt.boxplot([y1,y2]);
```



boxplot ?

- ref: https://github.com/mGalarnyk/Python_Tutorials/blob/master/Statistics/boxplot/box_plot.ipynb

```
np.random.seed(916170)

# connection path is here: https://stackoverflow.com/questions/6146290/plotting-a-line-ove
mu, sigma = 0, 1 # mean and standard deviation
s = np.random.normal(mu, sigma, 1000)

fig, axes = plt.subplots(nrows = 1, ncols = 1, figsize=(10, 5))

# rectangular box plot
bplot = axes.boxplot(s,
                vert=False,
                patch_artist=True,
                showfliers=True, # This would show outliers (the remaining .7% of the data
                positions = [0],
                boxprops = dict(linestyle='--', linewidth=2, color='Black', facecolor = 'r
                medianprops = dict(linestyle='-', linewidth=2, color='Yellow'),
                whiskerprops = dict(linestyle='-', linewidth=2, color='Blue', alpha = .4),
                capprops = dict(linestyle='-', linewidth=2, color='Black'),
                flierprops = dict(marker='o', markerfacecolor='green', markersize=10,
                  linestyle='none', alpha = .4),
                widths = .3,
                zorder = 1)

axes.set_xlim(-4, 4)
plt.xticks(fontsize = 14)

axes.set_yticks([])
axes.annotate(r'',
            xy=(-.73, .205), xycoords='data',
            xytext=(.66, .205), textcoords='data',
            arrowprops=dict(arrowstyle="|-|",
                            connectionstyle="arc3")
            );

axes.text(0, .25, "Interquartile Range \n(IQR)",  horizontalalignment='center', fontsize=1
axes.text(0, -.21, r"Median", horizontalalignment='center', fontsize=16);
axes.text(2.65, -.15, "\"Maximum\"", horizontalalignment='center', fontsize=18);
axes.text(-2.65, -.15, "\"Minimum\"", horizontalalignment='center', fontsize=18);
axes.text(-.68, -.24, r"Q1", horizontalalignment='center', fontsize=18);
axes.text(-2.65, -.21, r"(Q1 - 1.5*IQR)", horizontalalignment='center', fontsize=16);
axes.text(.6745, -.24, r"Q3", horizontalalignment='center', fontsize=18);
axes.text(.6745, -.30, r"(75th Percentile)", horizontalalignment='center', fontsize=12);
axes.text(-.68, -.30, r"(25th Percentile)", horizontalalignment='center', fontsize=12);
axes.text(2.65, -.21, r"(Q3 + 1.5*IQR)", horizontalalignment='center', fontsize=16);

axes.annotate('Outliers', xy=(2.93,0.015), xytext=(2.52,0.20), fontsize = 18,
            arrowprops={'arrowstyle': '->', 'color': 'black', 'lw': 2},
            va='center');

axes.annotate('Outliers', xy=(-3.01,0.015), xytext=(-3.41,0.20), fontsize = 18,
            arrowprops={'arrowstyle': '->', 'color': 'black', 'lw': 2},
            va='center');
```



plotly boxplot

- ( )

```
!pip install plotly
!pip install ipywidgets
```

```
!pip install jupyter-dash
!pip install dash
!pip install pandas
```

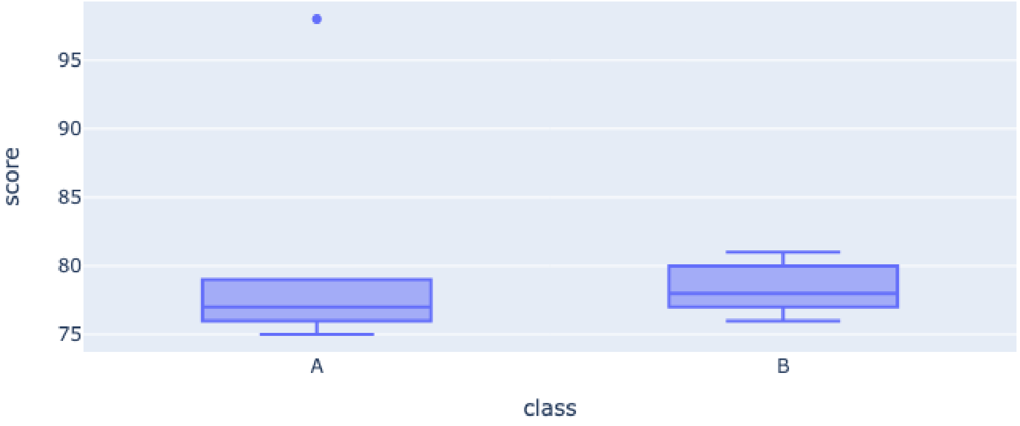
```python
import plotly.express as px
import pandas as pd
```

```python
df= pd.DataFrame({'score':y1+y2,'class':['A']*len(y1) + ['B']*len(y2)})
df
```

	score	class
0	75	A
1	75	A
2	76	A
3	76	A
4	77	A
5	77	A
6	79	A
7	79	A
8	79	A
9	98	A
10	76	B
11	76	B
12	77	B
13	77	B
14	78	B
15	78	B
16	80	B
17	80	B
18	80	B
19	81	B

```python
px.box(df,x='class',y='score')
```

```
Unable to display output for mime type(s): text/html
```



histogram

motivating example

- : A B ?

- .
- . , .

- “A B ?” . .

- B . A ! (? 2 3 , ?)

- “A B ?” .

- 1: .
- 2: .

“A B ?” .

(2) : ?

- A B .

```
np.random.seed(43052)
y1 = np.random.randn(10000)
```

```
y2 = np.random.randn(10000) + 0.5
```

```
np.mean(y1),np.mean(y2)
```

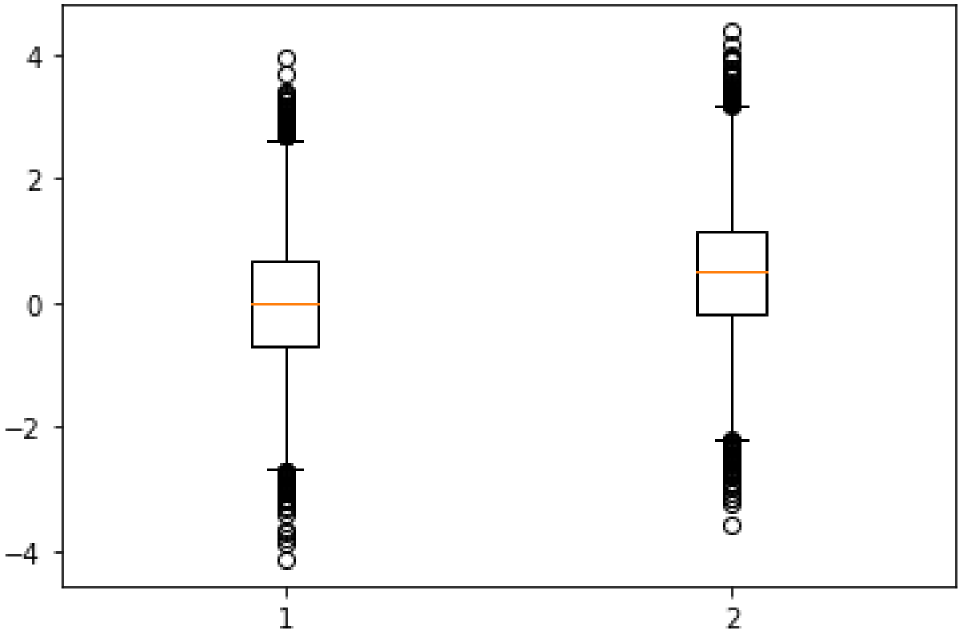
```
(-0.011790879905079434, 0.4979147460611458)
```

```
np.mean(y2) - np.mean(y1)
```

```
0.5097056259662253
```

y2 y1 0.5097056259662253 ?

```
plt.boxplot([y1,y2]);
```



- , +C 0.5 y !

- “B ≈ A + 0.5” . “A B ?” “B 0.5 ” .

- : 1,2

- ? (= ?): . ( )

histogram ?

- : X , Y

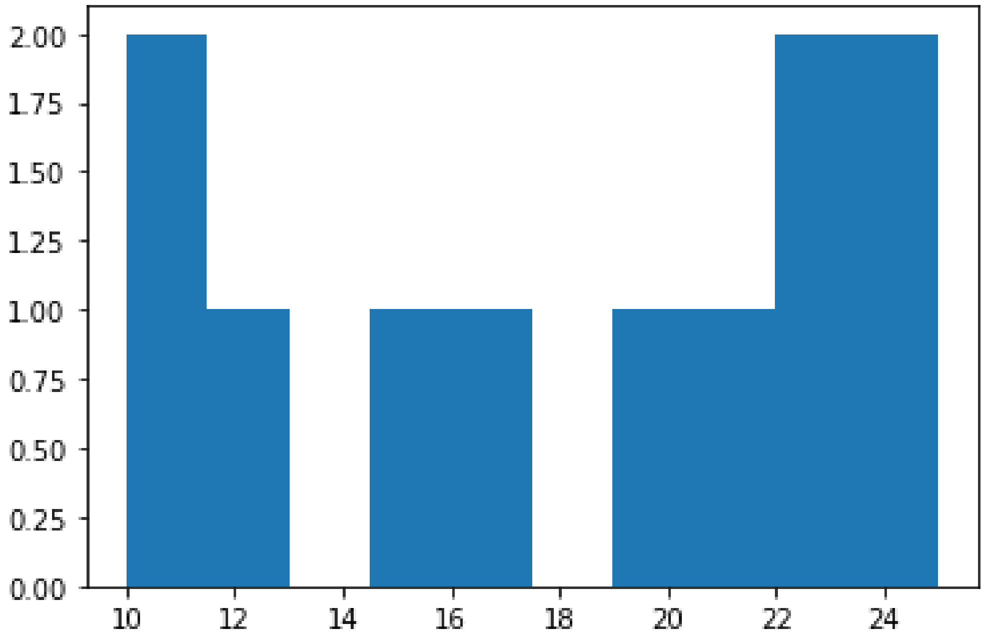
matplotlib histogram

- 1

```
y=[10,11,12,15,16,20,21,22,23,24,25]
```

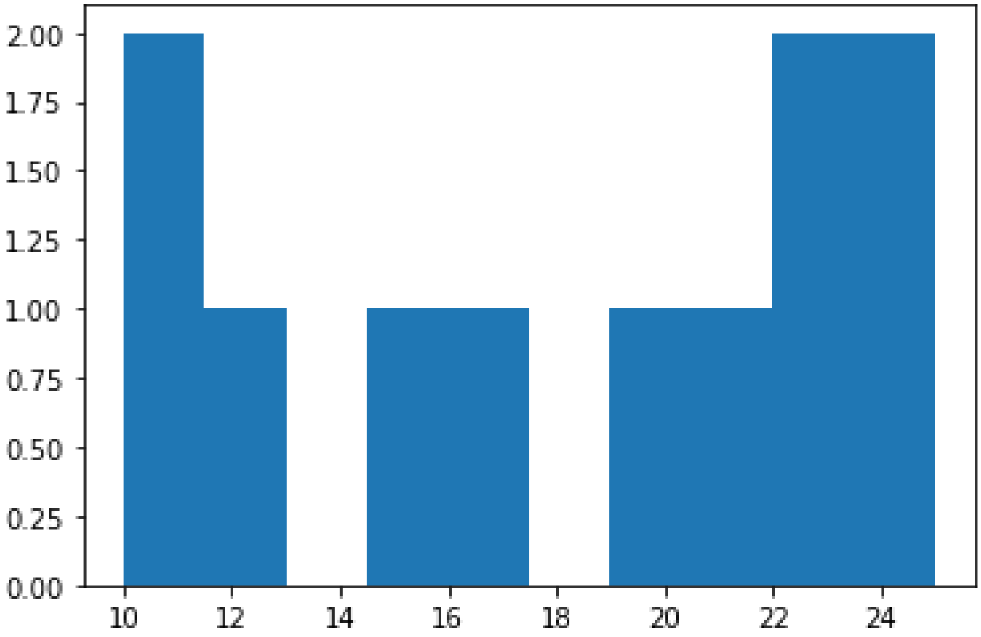
```
plt.hist(y)
```

```
(array([2., 1., 0., 1., 1., 0., 1., 1., 2., 2.]),
 array([10. , 11.5, 13. , 14.5, 16. , 17.5, 19. , 20.5, 22. , 23.5, 25. ]),
 <BarContainer object of 10 artists>)
```



```
plt.hist(y,bins=10)
```

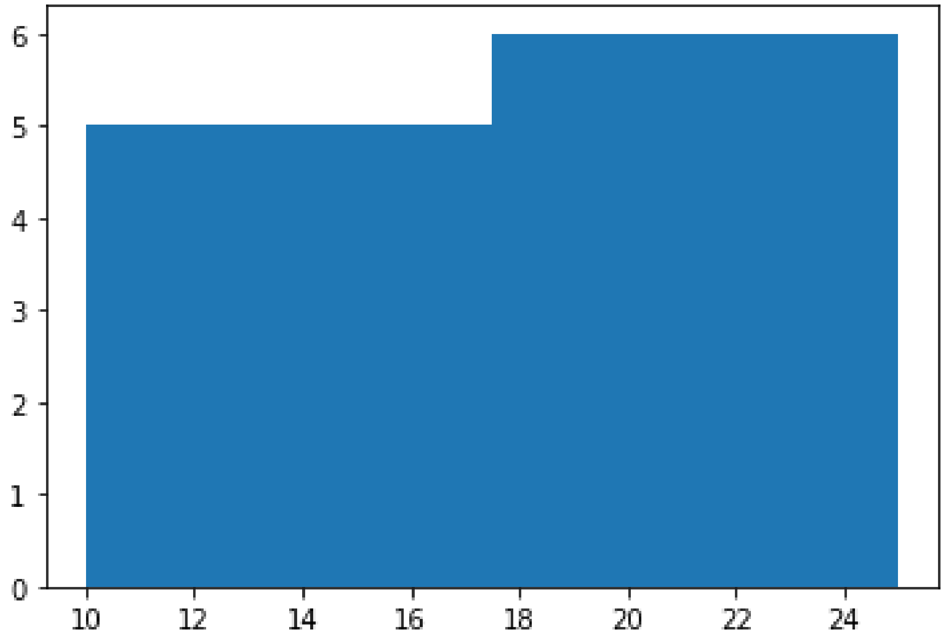
```
(array([2., 1., 0., 1., 1., 0., 1., 1., 2., 2.]),
 array([10. , 11.5, 13. , 14.5, 16. , 17.5, 19. , 20.5, 22. , 23.5, 25. ]),
 <BarContainer object of 10 artists>)
```



- 2

```
plt.hist(y,bins=2)
#plt.hist(y,bins=1)
```

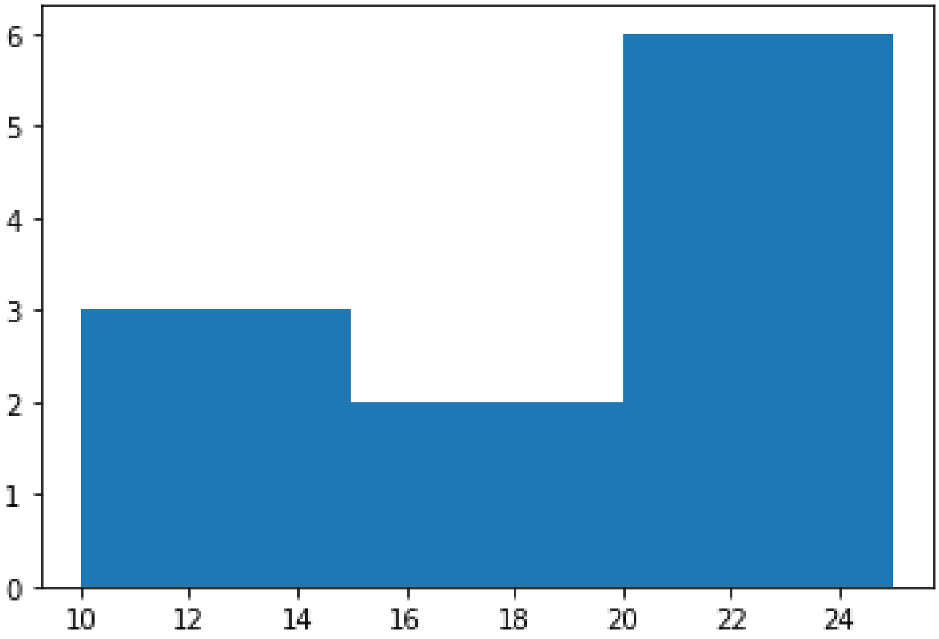
```
(array([5., 6.]),
 array([10. , 17.5, 25. ]),
 <BarContainer object of 2 artists>)
```



- 3

```
plt.hist(y,bins=3)
```

```
(array([3., 2., 6.]),
 array([10., 15., 20., 25.]),
 <BarContainer object of 3 artists>)
```

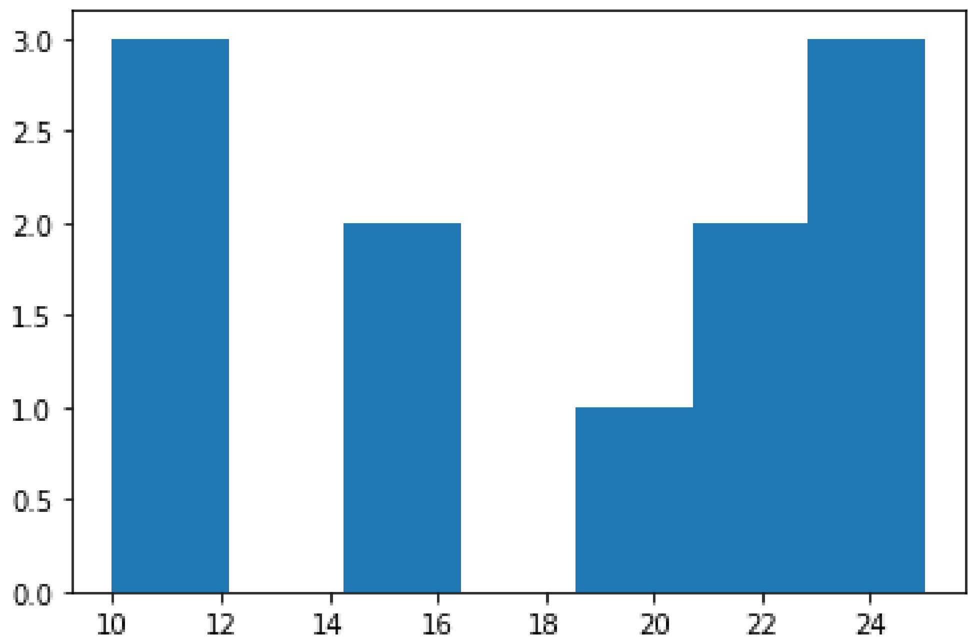


- 25, 10 range 15 .
- range / bins = 15 / 3 = 5 5 .
- [10,15), [15,20), [20,25] .
- 3,2,6 .

- 4

```
plt.hist(y,bins=7)
```

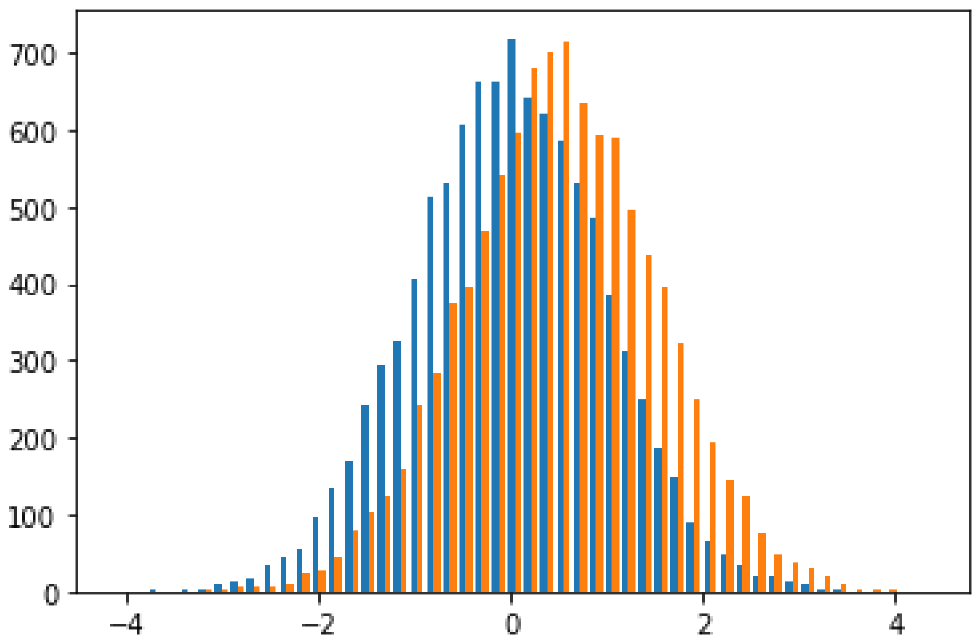
```
(array([3., 0., 2., 0., 1., 2., 3.]),
 array([10.        , 12.14285714, 14.28571429, 16.42857143, 18.57142857,
        20.71428571, 22.85714286, 25.        ]),
 <BarContainer object of 7 artists>)
```



- 25, 10 range 15 .
- range / bins = 15 / 7 = 2.142857142857143 2.142857142857143 .
- [10,12.14285714), [12.14285714,14.28571429,), [22.85714286,25] .
- 3,0,2,0,1,2,3 .

- 5

```
# np.random.seed(43052)
# y1 = np.random.randn(10000)
# y2 = np.random.randn(10000) + 0.5
plt.hist([y1,y2],bins=50);
```



seaborn histogram

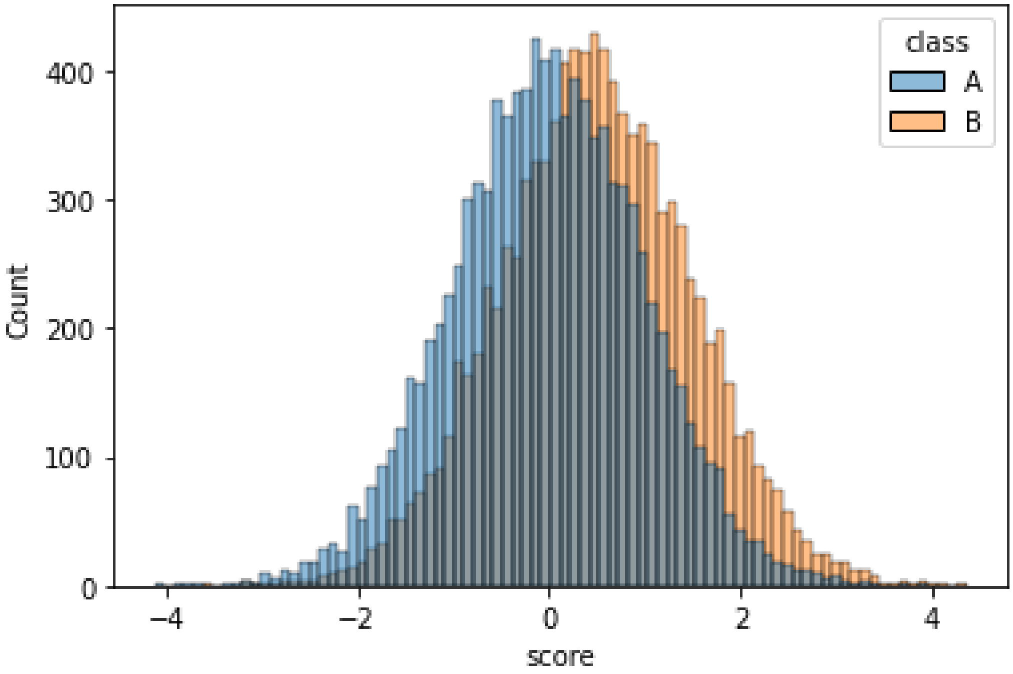
```
import seaborn as sns
```

```
df=pd.DataFrame({'score':np.concatenate([y1,y2]), 'class':['A']*len(y1)+['B']*len(y2)})
df
```

	score	class
0	0.383420	A
1	1.084175	A
2	1.142778	A
3	0.307894	A
4	0.237787	A
...	...	...
19995	0.493276	B
19996	0.619512	B
19997	-0.500529	B
19998	1.267551	B
19999	1.004863	B

```
sns.histplot(df,x='score',hue='class')
```

```
<AxesSubplot:xlabel='score', ylabel='Count'>
```

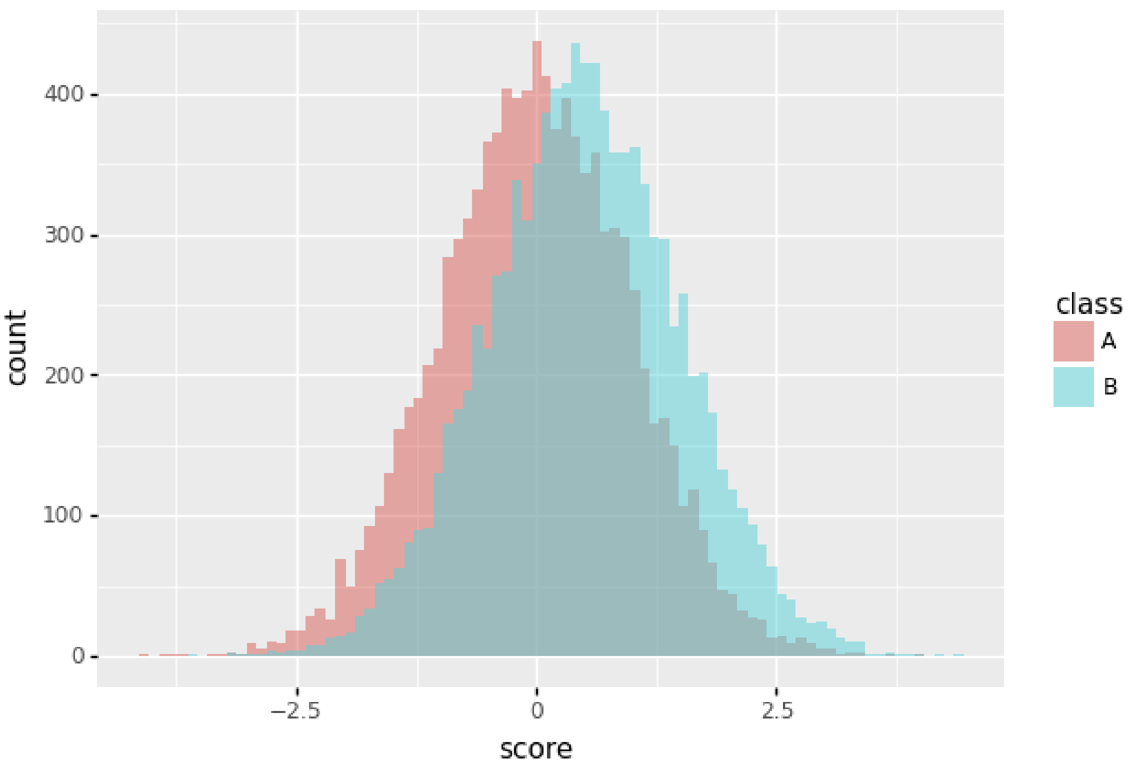


plotnine histogram

```
from plotnine import *
```

```
ggplot(df) + geom_histogram(aes(x='score',fill='class'),position='identity',alpha=0.5)
```

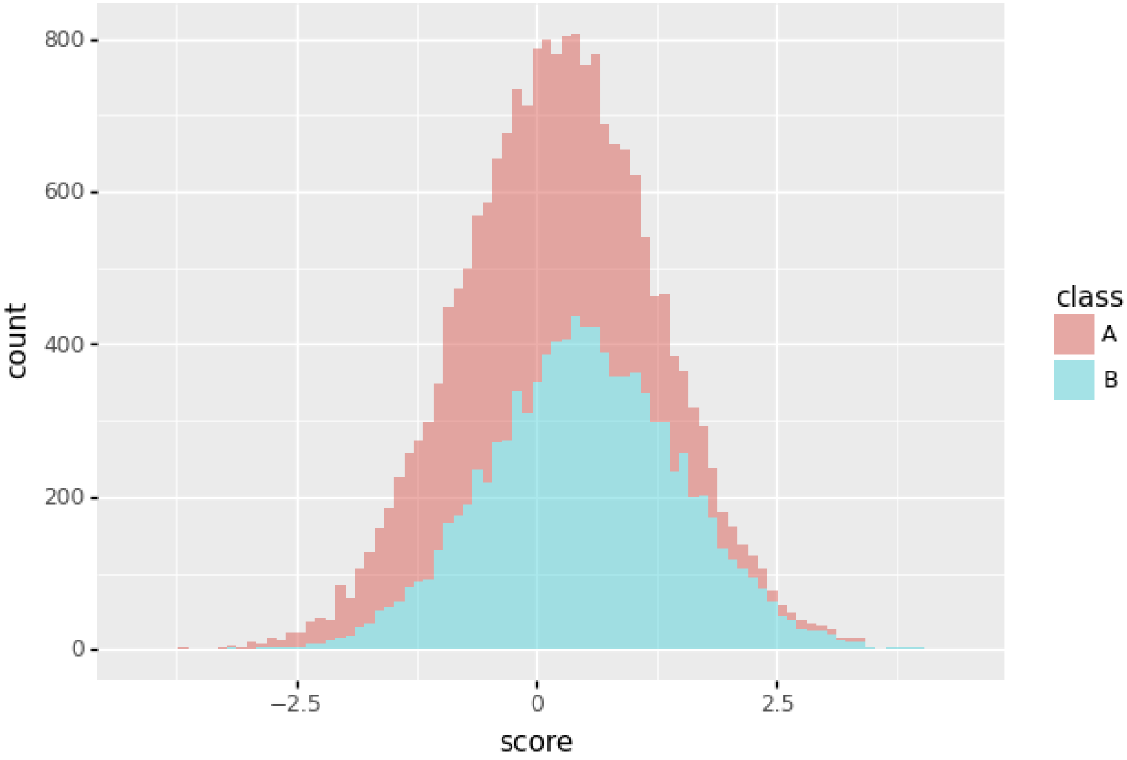
```
/home/cgb4/anaconda3/envs/py37/lib/python3.7/site-packages/plotnine/stats/stat_bin.py:95: Pl
```



```
<ggplot: (8787216362017)>
```

```
ggplot(df) + geom_histogram(aes(x='score',fill='class'),alpha=0.5) ##
```

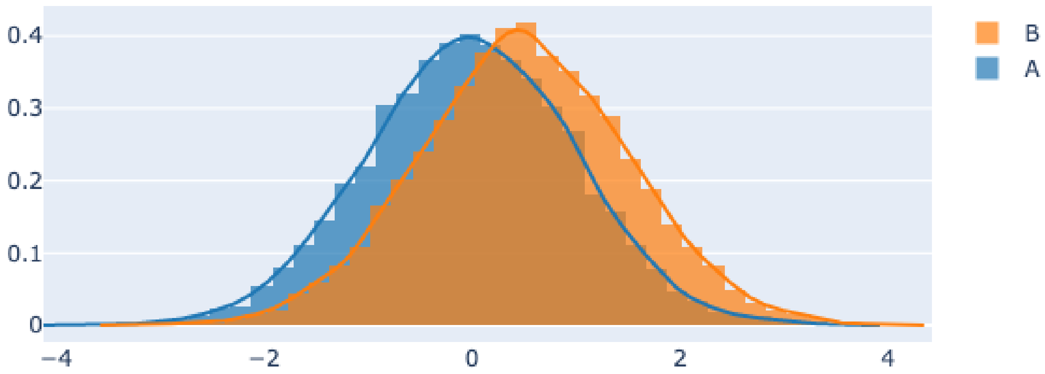
```
/home/cgb4/anaconda3/envs/py37/lib/python3.7/site-packages/plotnine/stats/stat_bin.py:95: Pl
```



```
<ggplot: (8787219120217)>
```

plotly histogram

```
import plotly.figure_factory as ff

hist_data = [y1, y2]

group_labels = ['A', 'B']

# Create distplot with curve_type set to 'normal'
ff.create_distplot(hist_data, group_labels,bin_size=.2, show_rug=False)
```



(1) np.random.seed(202043052)

(2) y1, y2 // 10

- y1: 0, =1
- y2: 1, =1

(3) plotly .